第一章 基础准备及入门

本章有两个目的：一是讲述 MATLAB 正常运行所必须具备的基础条件；二是简明系统地介绍高度集成的 Desktop 操作桌面的功能和使用方法。

本章的前两节分别讲述：MATLAB 的正确安装方法和 MATLAB 环境的启动。因为指令窗是 MATLAB 最重要的操作界面，所以本章用第 1.3、1.4 两节以最简单通俗的叙述、算例讲述指令窗的基本操作方法和规则。这部分内容几乎对 MATLAB 各种版本都适用。

MATLAB6.x 不同于其前版本的最突出之处是：向用户提供前所未有的、成系列的交互式工作界面。了解、熟悉和掌握这些交互界面的基本功能和操作方法，将使新老用户能事半功倍地利用 MATLAB 去完成各种学习和研究。为此，本章特设几节用于专门介绍最常用的交互界面：历史指令窗、当前目录浏览器、工作空间浏览器、内存数组编辑器、交互界面分类目录窗、M 文件编辑/调试器、及帮助导航/浏览器。

本章是根据 MATLAB6.5 版编写的，但大部分内容也适用于其他 6.x 版。

1.1MATLAB 的安装和内容选择

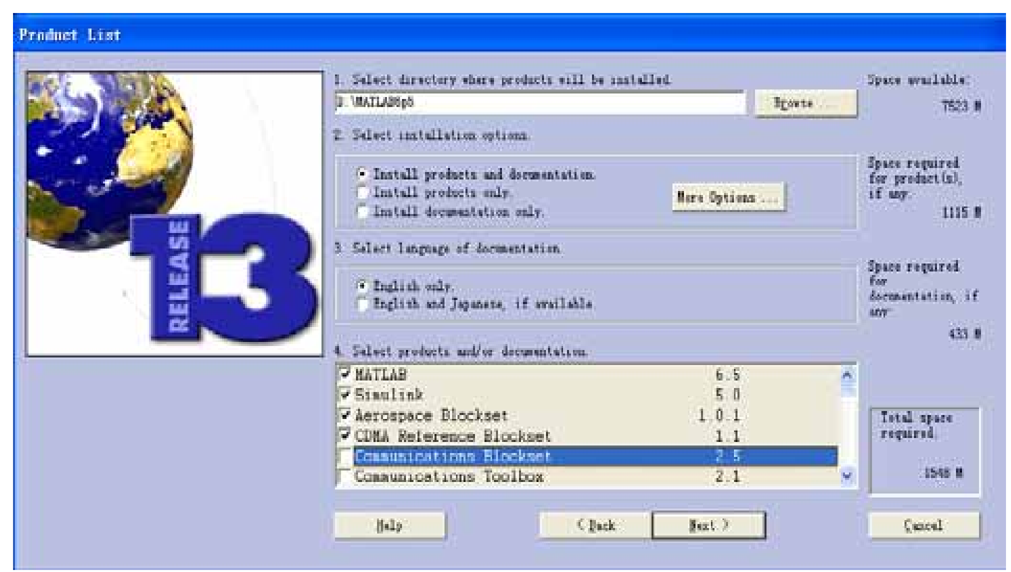


图 1.1-1

1.2Desktop 操作桌面的启动

1.2.1 MATLAB 的启动

1.2.2 Desktop 操作桌面简介

一 操作桌面的缺省外貌

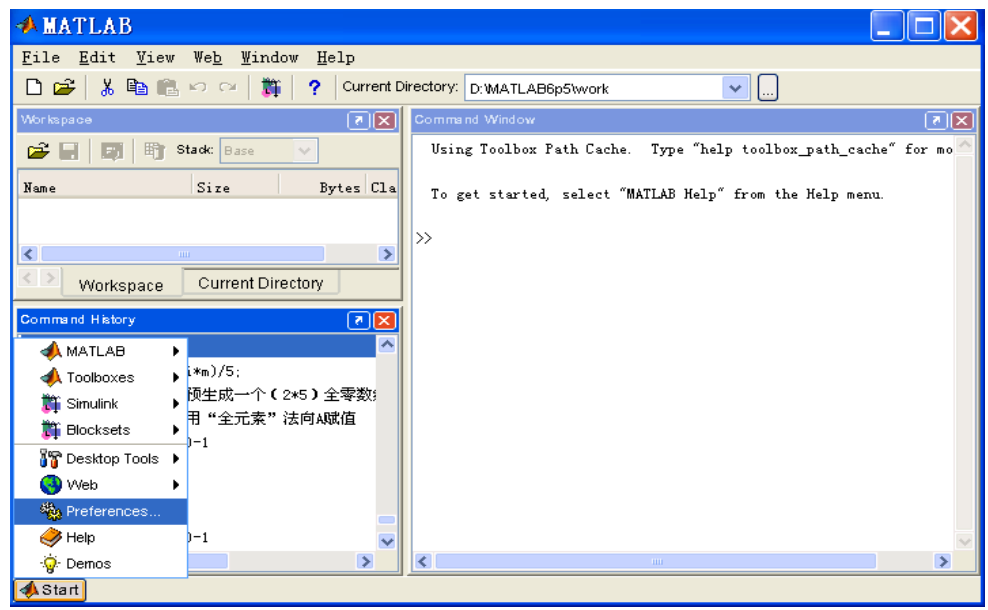


图 1.2-1

二 通用操作界面

1.3Command Window 运行入门

1.3.1 Command Window 指令窗简介

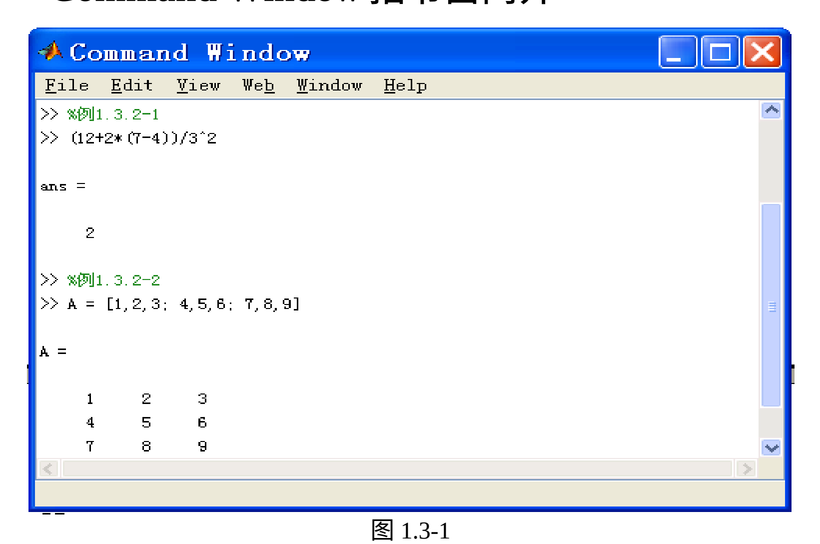


图 1.3-1

1.3.2 最简单的计算器使用法

【例 1.3.2-1】求$[12+2\times(7-4)]\div3^2$的算术运算结果。

（1）用键盘在 MATLAB 指令窗中输入以下内容

```
>> (12+2*(7-4))/3^2
```

（2）在上述表达式输入完成后，按【Enter】键，该就指令被执行。

（3）在指令执行后，MATLAB 指令窗中将显示以下结果。

```
ans =
     2
```

【例 1.3.2-2】简单矩阵 $A=\begin{bmatrix}1&2&3\\4&5&6\\7&8&9\end{bmatrix}$ 的输入步骤。

（1）在键盘上输入下列内容

```
A = [1,2,3; 4,5,6; 7,8,9]
```

（2）按【Enter】键，指令被执行。

（3）在指令执行后，MATLAB 指令窗中将显示以下结果：

```
A =
     1     2     3
     4     5     6
     7     8     9
```

【例 1.3.2-3】矩阵的分行输入。

```
A=[1,2,3
   4,5,6
   7,8,9]
A =
     1     2     3
     4     5     6
     7     8     9
```

【例 1.3.2-4】指令的续行输入

```
S=1-1/2+1/3-1/4+ ...
1/5-1/6+1/7-1/8
S =
    0.6345
```

1.3.3　　数值、变量和表达式

一 数值的记述

二 变量命名规则

三 MATLAB 默认的预定义变量

四 运算符和表达式

五 复数和复数矩阵

【例 1.3.3-1】复数 $z_1=3+4i, z_2=1+2i, z_3=2e^{\frac{\pi}{6}i}$ 表达，及计算 $z=\frac{z_1 z_2}{z_3}$。

（1）

```
z1= 3 + 4i
z1 =
   3.0000 + 4.0000i
```

（2）

```
z2 = 1 + 2 * i
z3=2*exp(i*pi/6)
z=z1*z2/z3
z2 =
   1.0000 + 2.0000i
z3 =
   1.7321 + 1.0000i
z =
   0.3349 + 5.5801i
```

【例 1.3.3-2】 复数矩阵的生成及运算

```
A=[1,3;2,4]-[5,8;6,9]*i
B=[1+5i,2+6i;3+8*i,4+9*i]
C=A*B
A =
   1.0000 - 5.0000i   3.0000 - 8.0000i
   2.0000 - 6.0000i   4.0000 - 9.0000i
B =
   1.0000 + 5.0000i   2.0000 + 6.0000i
   3.0000 + 8.0000i   4.0000 + 9.0000i
C =
  1.0e+002 *
   0.9900             1.1600 - 0.0900i
   1.1600 + 0.0900i   1.3700
```

【例 1.3.3-3】求上例复数矩阵 C 的实部、虚部、模和相角。

```
C_real=real(C)
C_imag=imag(C)
C_magnitude=abs(C)
C_phase=angle(C)*180/pi         %以度为单位计算相角
C_real =
    99   116
   116   137
C_imag =
     0    -9
     9     0
C_magnitude =
   99.0000  116.3486
  116.3486  137.0000
C_phase =
         0   -4.4365
    4.4365         0
```

【例 1.3.3-4】 用 MATLAB 计算 $\sqrt[3]{-8}$ 能得到 –2 吗？

（1）

```
a=-8;
r=a^(1/3)
r =
   1.0000 + 1.7321i
```

（2）

```
m=[0,1,2];
R=abs(a)^(1/3);
Theta=(angle(a)+2*pi*m)/3;
rrr=R*exp(i*Theta)
rrr =
   1.0000 + 1.7321i  -2.0000 + 0.0000i   1.0000 - 1.7321i
```

（3）

```
t=0:pi/20:2*pi;x=R*sin(t);y=R*cos(t);
plot(x,y,'b:'),grid
hold on
plot(rrr(1),'.','MarkerSize',50,'Color','r')
plot(rrr([2,3]),'o','MarkerSize',15,'Color','b')
axis([-3,3,-3,3]),axis square
hold off
```

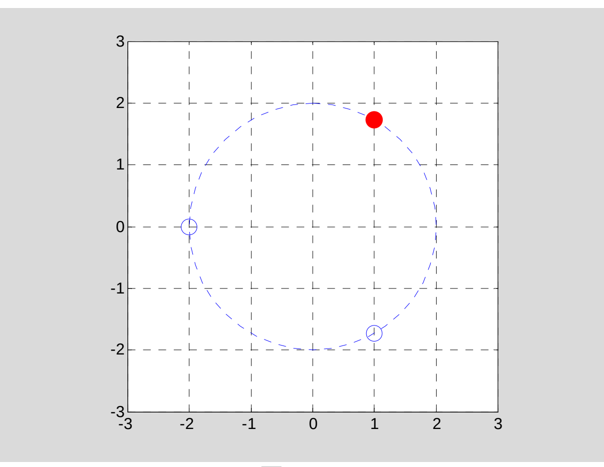


图 1.3-2

1.3.4 计算结果的图形表示

【例 1.3.4-1】画出衰减振荡曲线 $y=e^{-\frac{t}{3}}\sin 3t$ 及其它的包络线 $y_0=e^{-\frac{t}{3}}$。t 的取值范围是 $[0,4\pi]$。（图 1.3-3）

```
t=0:pi/50:4*pi;
y0=exp(-t/3);
y=exp(-t/3).*sin(3*t);
plot(t,y,'-r',t,y0,':b',t,-y0,':b')
grid
```

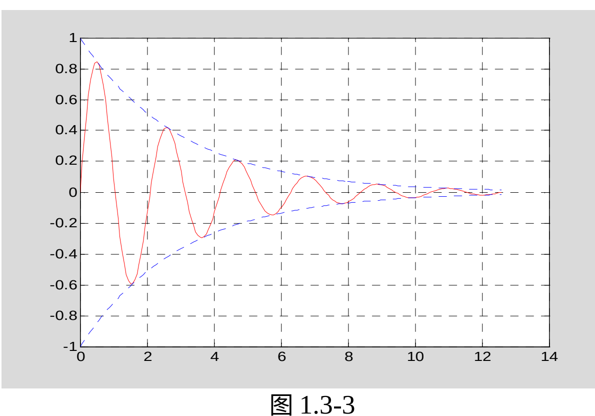


图 1.3-3

【例 1.3.4-2】画出 $z=\frac{\sin(\sqrt{x^2+y^2})}{\sqrt{x^2+y^2}}$ 所表示的三维曲面（图 1.3-4）。x,y 的取值范围是 $[-8,8]$。

```
clear;x=-8:0.5:8;
y=x';
X=ones(size(y))*x;
Y=y*ones(size(x));
R=sqrt(X.^2+Y.^2)+eps;    %<5>
Z=sin(R)./R;              %<6>
surf(X,Y,Z);              %
```

```
colormap(cool)            %
xlabel('x'),ylabel('y'),zlabel('z')
```

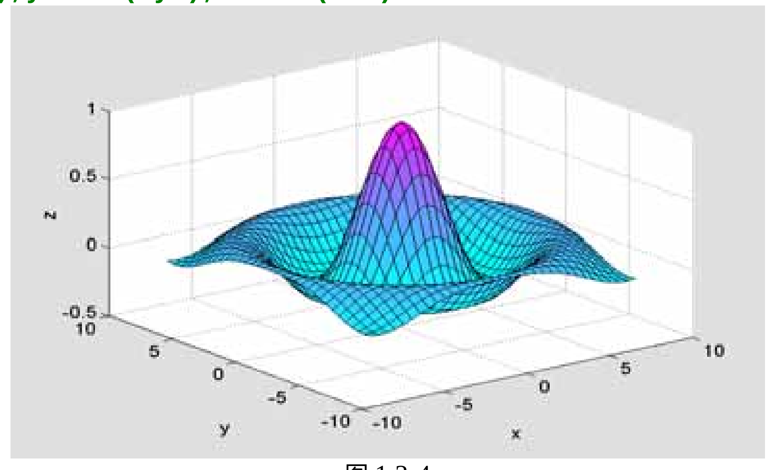


图 1.3-4

1.4Command Window 操作要旨

1.4.1　　　指令窗显示方式的操作

一 缺省显示方式

二 显示方式的设置

1.4.2　　　数值计算结果的显示格式

1.4.3　　　指令行中的标点符号

1.4.4　　　指令窗的常用控制指令

1.4.5　　　指令窗中指令行的编辑

【例 1.4.5-1】指令行操作过程示例。

1.5Command History 和实录指令 diary

1.5.1　　　Command History 历史指令窗简介



图 1.5-1

1.5.2 历史指令行的再运行

【例 1.5.2-1】再运行图 1.5-2 所示历史指令窗中的三行指令。

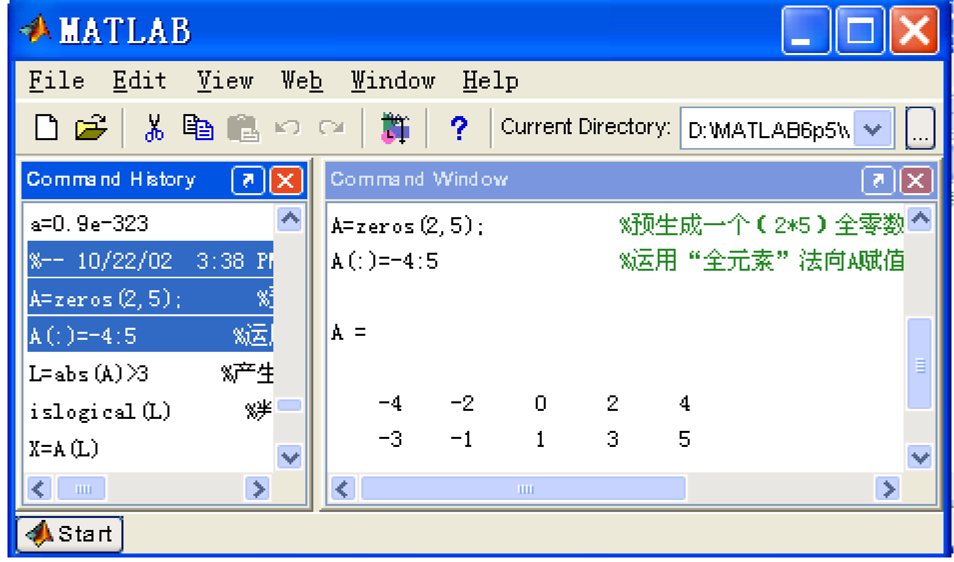


图 1.5-2

1.5.3 指令窗实录指令 diary

1.6 Current Directory、路径设置器和文件管理

1.6.1 Current Directory 当前目录浏览器简介

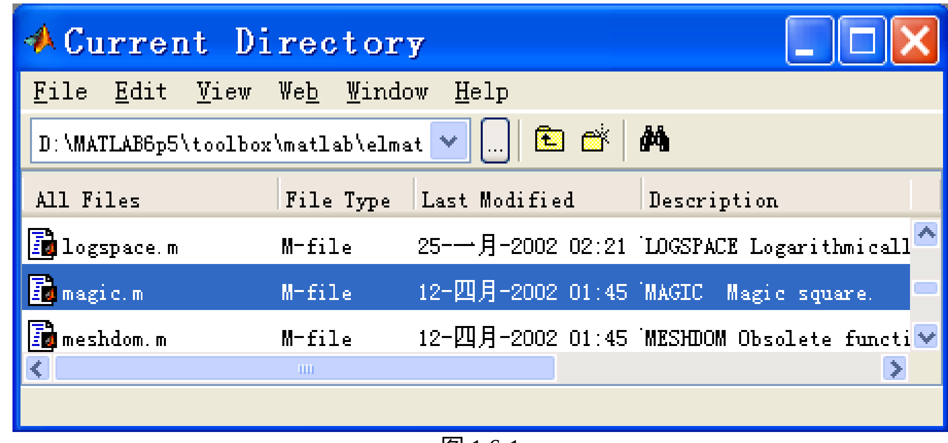


图 1.6-1

一 用户目录和当前目录设置

二 借助当前目录浏览器获取 M 和 MAT 文件信息

【例 1.6.1-1】从图 1.2-1 所示 MATLAB 缺省桌面开始，叙述引出图 1.6-1 所示面貌的当前目录浏览器的操作过程。

1.6.2 MATLAB 的搜索路径

1.6.3 MATLAB 搜索路径的扩展和修改

一 何时需要修改搜索路径

二 利用设置路径对话框修改搜索路径

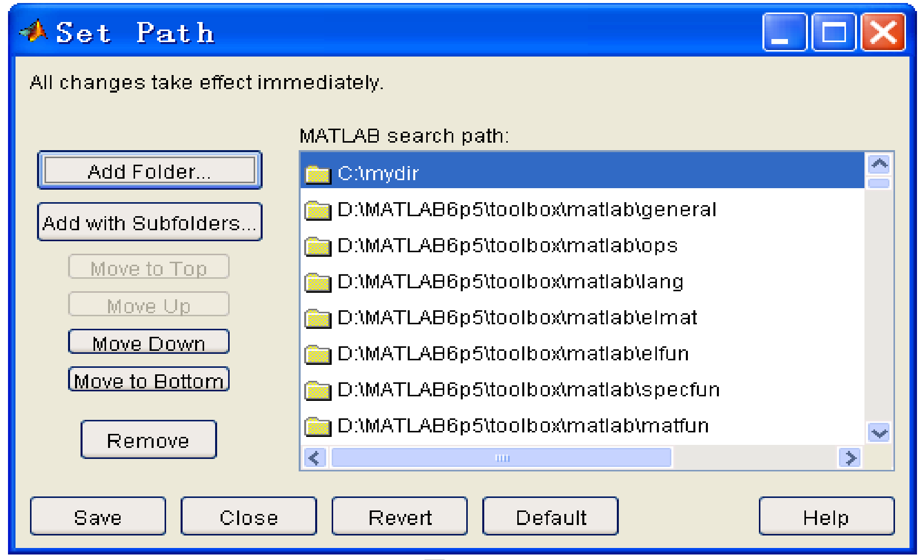


图 1.6-2

三 利用指令 path 设置路径

1.7 Workspace Browser 和 Array Editor

1.7.1 Workspace Browser 工作空间浏览器简介

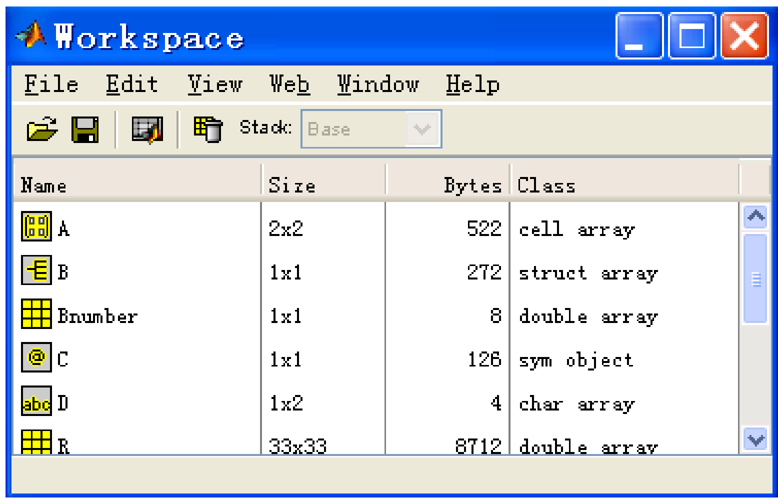


图 1.7-1

1.7.2 现场菜单用于内存变量的查阅和删除

一 内存变量查阅、删除的指令操作法

【例 1.7.2-1】在指令窗中运用 who, whos 查阅 MATLAB 内存变量。

```
who
Your variables are:

A          Bnumber   D         R         XYZ       Z         y
B          C         DD        X         Y         x

whos
  Name              Size           Bytes  Class

  A                 2x2              230  cell array
  B                 1x1              264  struct array
  Bnumber           1x1                8  double array
  C                 2x2              408  sym object
  D                 1x2                4  char array
  DD                2x2                8  char array
  R               33x33             8712  double array
  X               33x33             8712  double array
  XYZ             33x33x3          26136  double array
  Y               33x33             8712  double array
  Z               33x33             8712  double array
  x                1x33              264  double array
  y               33x1               264  double array

Grand total is 7722 elements using 62434 bytes
```

【例 1.7.2-2】在指令窗中运用 clear 指令可以删除内存中的变量。

```
clear Bnumber
who
Your variables are:

A    B    C    D    DD   R    X    XYZ  Y    Z    x    y
```

二 内存变量查阅和删除的现场菜单操作法

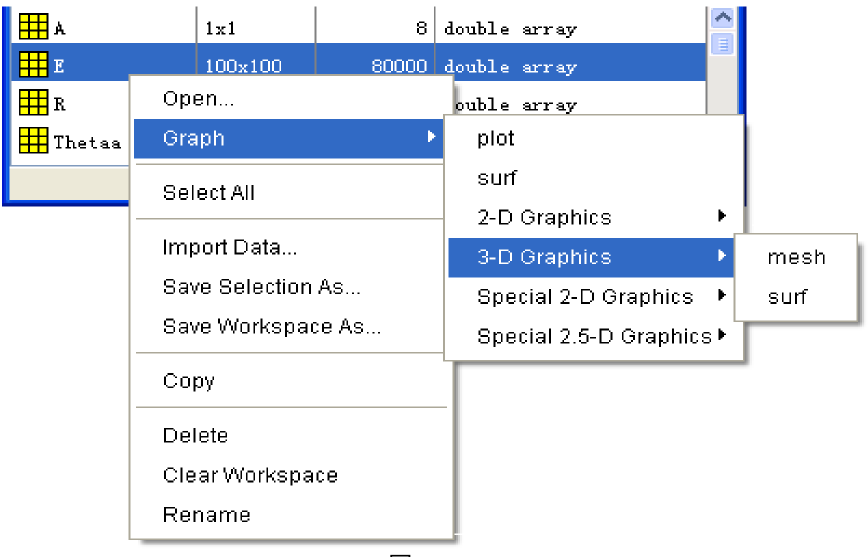


图 1.7-2

【例 1.7.2-3】通过 “ 工作空间浏览器 ” 的运作，采用图形显示内存变量 Z 。

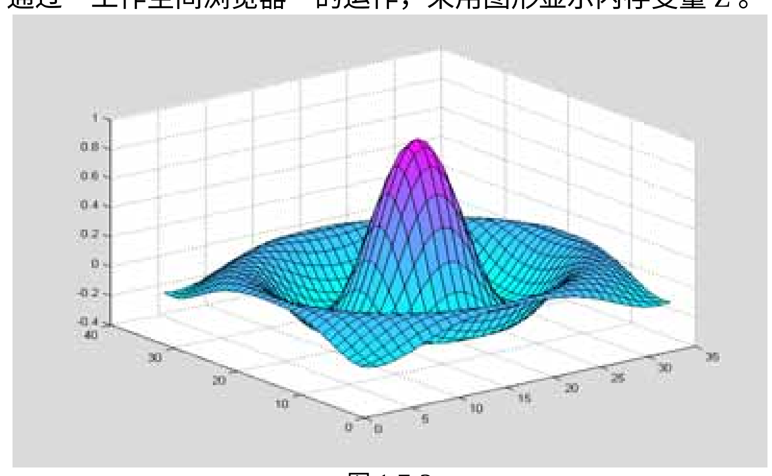


图 1.7-3

【例 1.7.2-4】通过 “ 工作空间浏览器 ” 删除内存变量。

1.7.3 Array Editor 数组编辑器和大数组的输入

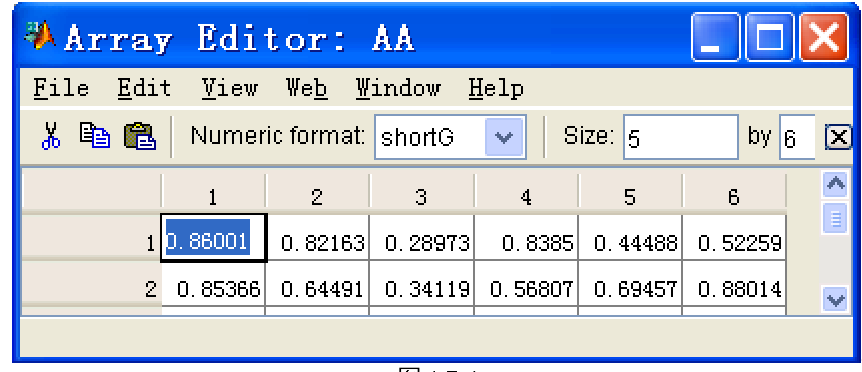


图 1.7-4

1.7.4 数据文件的存取

一 存取数据文件的指令操作法

二 通过内存变量浏览器实现数据文件的存取

（1） 产生保存全部内存变量的数据文件的操作方法

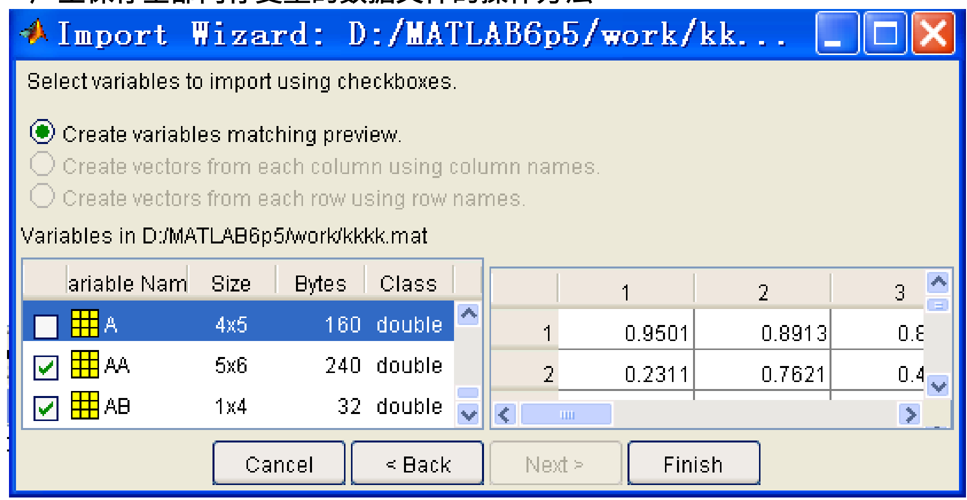


图 1.7-5

【例 1.7.4-1】数据的存取。（假定内存中已经存在变量 X,Y,Z）

（1）

```
mkdir('c:\','my_dir');
cd c:\my_dir
save  saf X Y Z
dir
.        ..       saf.mat
```

（2）

```
clear
load  saf  Z
who
Your variables are:
Z
```

1.8Launch Pad 交互界面分类目录窗

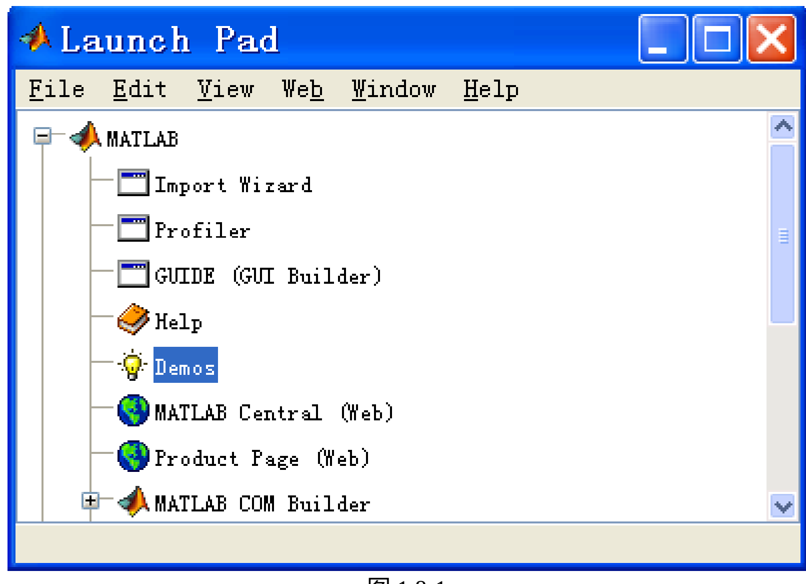


图 1.8-1

1.9Editor/Debugger 和脚本编写初步

1.9.1 Editor/Debugger M 文件编辑调试器简介

一 编辑调试器的开启

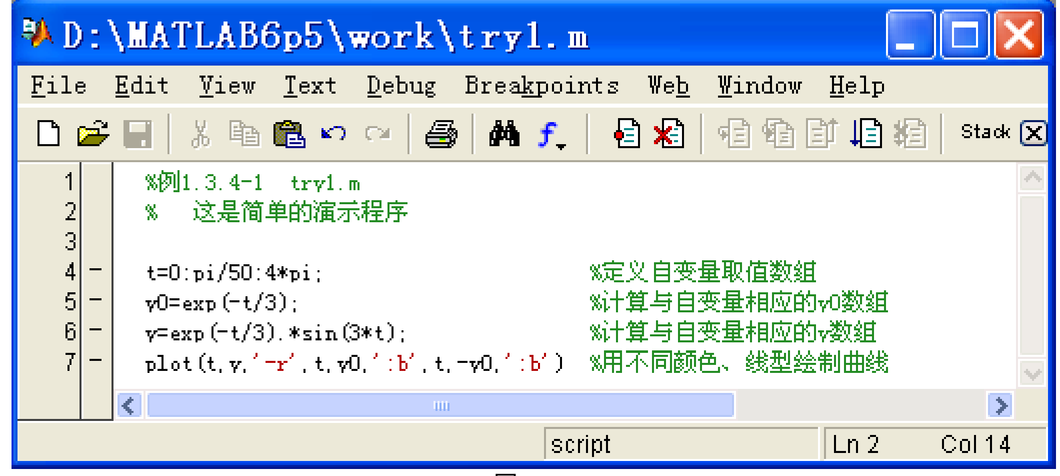


图 1.9-1

二 编辑器使用中的若干注意事项

1.9.2 M 脚本文件编写初步

【例 1.9.2-1】编写解算例 1.3.4-1 题目的 M 脚本文件，并运行之。

操作步骤：

1.10 帮助系统

1.10.1 帮助方式概述

一 “ 纯文本 ” 帮助

【例 1.10.1-1】在指令窗中运行 help 的示例。

(1)

```
help help
 HELP On-line help, display text at command line.
    HELP, by itself, lists all primary help topics.  Each primary topic
    corresponds to a directory name on the MATLABPATH.
 ......     ......      ......
```

(2)

```
help
HELP topics:
matlab\general        -  General purpose commands.
matlab\ops            -  Operators and special characters.
matlab\lang           -  Programming language constructs.
matlab\elmat          -  Elementary matrices and matrix manipulation.
matlab\elfun          -  Elementary math functions.
 ......     ......      ......
For more help on directory/topic, type "help topic".
```

(3)

```
help elmat
Elementary matrices and matrix manipulation.
   Elementary matrices.
    zeros        - Zeros array.
    ones         - Ones array.
    eye          - Identity matrix.
 ......     ......      ......
```

(4)

```
help eye
EYE Identity matrix.
    EYE(N) is the N-by-N identity matrix.
    EYE(M,N) or EYE([M,N]) is an M-by-N matrix with 1's on
    the diagonal and zeros elsewhere.
    EYE(SIZE(A)) is the same size as A.
    See also ONES, ZEROS, RAND, RANDN.
```

【例 1.10.1-2】在指令窗中，运用 lookfor 找 H1 行（M 函数文件的第一注释行）

```
lookfor fourier
FFT Discrete Fourier transform.
FFT2 Two-dimensional discrete Fourier Transform.
FFTN N-dimensional discrete Fourier Transform.
IFFT Inverse discrete Fourier transform.
IFFT2 Two-dimensional inverse discrete Fourier transform.
IFFTN N-dimensional inverse discrete Fourier transform.
XFOURIER Graphics demo of Fourier series expansion.
MOT563_FFT Discrete Fourier transform.
MOT563_IFFT Inverse discrete Fourier transform.
MOT566_FFT Discrete Fourier transform.
MOT566_IFFT Inverse discrete Fourier transform.
DFTMTX Discrete Fourier transform matrix.
INSTDFFT Inverse non-standard 1-D fast Fourier transform.
NSTDFFT Non-standard 1-D fast Fourier transform.
FFT  Quantized Fast Fourier Transform.
FOURIER Fourier integral transform.
IFOURIER Inverse Fourier integral transform.
```

二 “ 导航/浏览器交互界面 ” 帮助

三 PDF 帮助

四 其他帮助

1.10.2 Help Navigator/Browser 帮助导航/浏览器简介

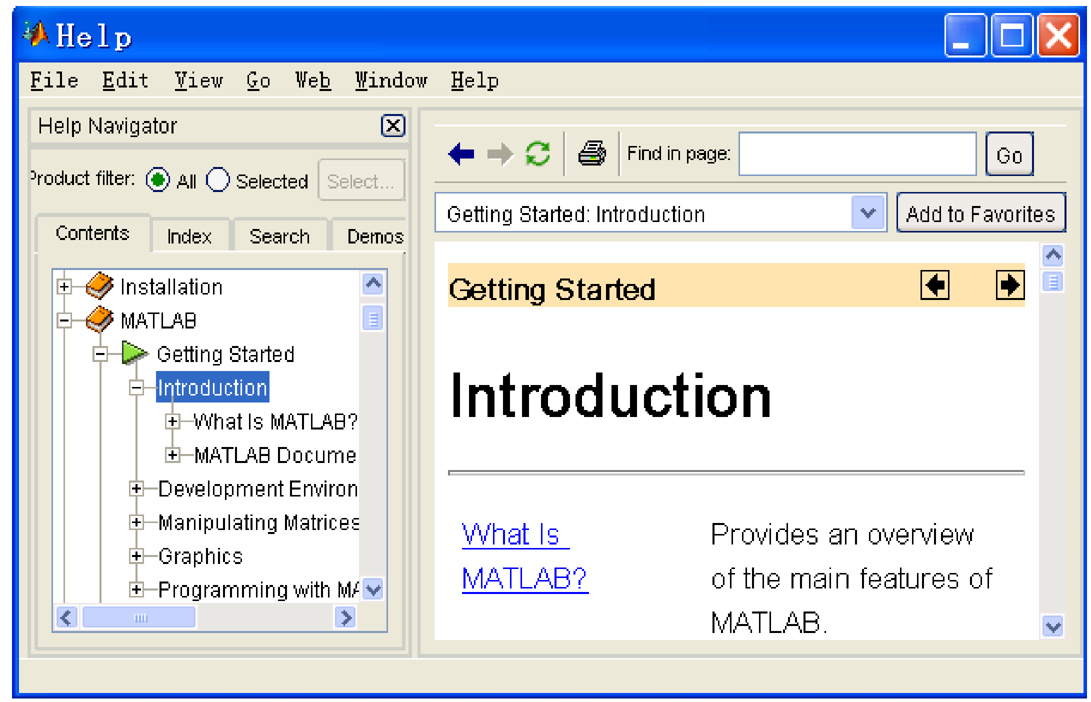


图 1.10-1

一 Contents 帮助文件目录窗

【例 1.10.2-1】通过鼠标操作获得如图 1.10-1 所示的界面。

二 Index 帮助索引窗

【例 1.10.2-2】利用 Idex 搜索 fourier 这条术语。（注意把本例与例 1.10.1-2、例 1.10.2-3 比较。）

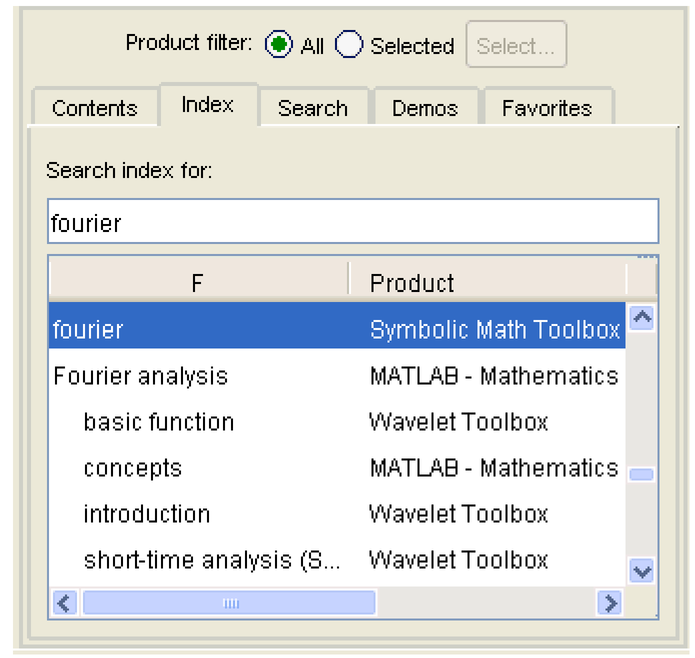


图 1.10-2

三 Search 搜索窗

【例 1.10.2-3】利用 " Search " 窗搜索词汇 fourier。（注意把本例与例 1.10.1-2、例 1.10.2-2 比较。）

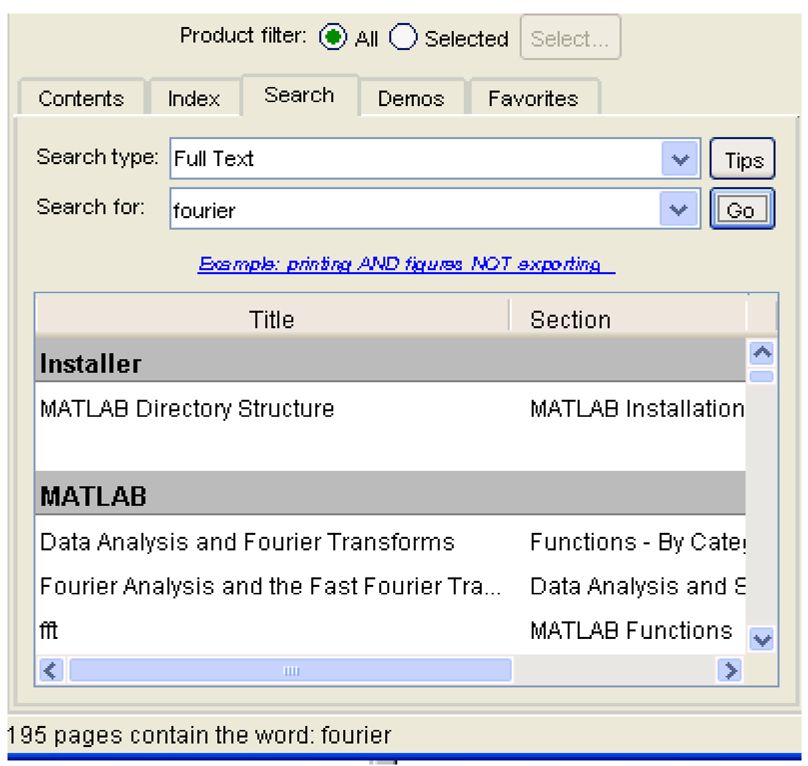


图 1.10-3

四 Favorites 书签窗

第二章　数值数组及其运算

数值数组（Numeric Array）和数组运算（Array Operations）始终是 MATLAB 的核心内容。自 MATLAB5.x 版起，由于其“面向对象”的特征，这种数值数组（以下简称为数组）成为了 MATALB 最重要的一种内建数据类型（Built-in Data Type），而数组运算就是定义在这种数据结构上的方法（Method）。

本章系统阐述：一、二维数值数组的创建、寻访；数组运算和矩阵运算的区别；实现数组运算的基本函数；多项式的表达、创建和操作；常用标准数组生成函数和数组构作技法；高维数组的创建、寻访和操作；非数 NaN、“空”数组概念和应用；关系和逻辑操作。

顺便指出：（1）本章所涉内容和方法，不仅使用于数值数组，而且也将部分地延伸使用于在其他数据结构中。（2）MATLAB5.x 和 6.x 版在本章内容上的差异极微。（3）MATLAB6.5 版新增的两种逻辑操作，在第 2.13.2 节给予介绍。

2.1　　引导

【例 2.1-1】绘制函数 $y = xe^{-x}$ 在 $0 \le x \le 1$ 时的曲线。

```
x=0:0.1:1
y=x.*exp(-x)
plot(x,y),xlabel('x'),ylabel('y'),title('y=x*exp(-x)')
x =
  Columns 1 through 7
         0    0.1000    0.2000    0.3000    0.4000    0.5000    0.6000
  Columns 8 through 11
    0.7000    0.8000    0.9000    1.0000
y =
  Columns 1 through 7
         0    0.0905    0.1637    0.2222    0.2681    0.3033    0.3293
  Columns 8 through 11
    0.3476    0.3595    0.3659    0.3679
```

图 2.1-1

2.2　　一维数组的创建和寻访

2.2.1　　一维数组的创建

2.2.2　　一维数组的子数组寻访和赋值

【例 2.2.2-1】子数组的寻访（Address）。

```
rand('state',0)
x=rand(1,5)
x =
    0.9501    0.2311    0.6068    0.4860    0.8913

x(3)
ans =
    0.6068

x([1 2 5])
ans =
    0.9501    0.2311    0.8913

x(1:3)
ans =
    0.9501    0.2311    0.6068

x(3:end)      %
ans =
    0.6068    0.4860    0.8913

x(3:-1:1)     %
ans =
    0.6068    0.2311    0.9501

x(find(x>0.5))
ans =
    0.9501    0.6068    0.8913

x([1 2 3 4 4 3 2 1])
ans =
  Columns 1 through 7
    0.9501    0.2311    0.6068    0.4860    0.4860    0.6068    0.2311
  Column 8
    0.9501
```

【例 2.2.2-2】子数组的赋值（Assign）。

```
x(3) = 0
x =
    0.9501    0.2311         0    0.4860    0.8913

x([1 4])=[1 1]
x =
    1.0000    0.2311         0    1.0000    0.8913
```

2.3 二维数组的创建

2.3.1 直接输入法

【例 2.3.1-1】在 MATLAB 环境下，用下面三条指令创建二维数组 C。

```
a=2.7358; b=33/79;
C=[1,2*a+i*b,b*sqrt(a);sin(pi/4),a+5*b,3.5+i]
C =
   1.0000             5.4716 + 0.4177i   0.6909
   0.7071             4.8244             3.5000 + 1.0000i
```

【例 2.3.1-2】复数数组的另一种输入方式。

```
M_r=[1,2,3;4,5,6],M_i=[11,12,13;14,15,16]
CN=M_r+i*M_i
M_r =
     1     2     3
     4     5     6
M_i =
    11    12    13
    14    15    16
CN =
   1.0000 +11.0000i   2.0000 +12.0000i   3.0000 +13.0000i
   4.0000 +14.0000i   5.0000 +15.0000i   6.0000 +16.0000i
```

2.3.2 利用 M 文件创建和保存数组

【例 2.3.2-1】创建和保存数组 AM 的 MyMatrix.m 文件。

（1）

```
% MyMatrix.m     Creation and preservation of matrix AM
AM=[101,102,103,104,105,106,107,108,109;...
    201,202,203,204,205,206,207,208,209;...
    301,302,303,304,305,306,307,308,309];
```

（2）

（3）

2.4 二维数组元素的标识

2.4.1 “ 全下标 ” 标识

2.4.2 “ 单下标 ” 标识

2.4.3 “ 逻辑 1 ” 标识

【例 2.4.3-1】找出数组 $A=\begin{bmatrix}-4 & -2 & 0 & 2 & 4\\ -3 & -1 & 1 & 3 & 5\end{bmatrix}$ 中所有绝对值大于 3 的元素。

```
A=zeros(2,5);
A(:)=-4:5
L=abs(A)>3
islogical(L)
X=A(L)
A =
    -4    -2     0     2     4
    -3    -1     1     3     5
L =
     1     0     0     0     1
     0     0     0     0     1
ans =
     1
X =
    -4
     4
     5
```

【例 2.4.3-2】演示逻辑数组与一般双精度数值数组的关系和区别。（本例在例 2.4.3-1 基础

上进行）。

（1）

```
Num=[1,0,0,0,1;0,0,0,0,1];
N_L=Num==L
c_N=class(Num)
c_L=class(L)
N_L =
     1     1     1     1     1
     1     1     1     1     1
c_N =
double
c_L =
double
```

（2）

```
islogical(Num)
Y=A(Num)
ans =
     0
??? Index into matrix is negative or zero.  See release notes on changes
to
logical indices.
```

2.5 二维数组的子数组寻访和赋值

【例 2.5-1】不同赋值方式示例。

```
A=zeros(2,4)
A =
     0     0     0     0
     0     0     0     0

A(:)=1:8
A =
     1     3     5     7
     2     4     6     8

s=[2 3 5];
A(s)
Sa=[10 20 30]'
A(s)=Sa
ans =
     2     3     5
Sa =
    10
    20
    30
A =
     1    20    30     7
    10     4     6     8
A(:,[2 3])=ones(2)
A =
     1     1     1     7
    10     1     1     8
```

2.6 执行数组运算的常用函数

2.6.1 函数数组运算规则的定义：

2.6.2 执行数组运算的常用函数

【例 2.6.2-1】演示 pow2 的数组运算性质。

```
A=[1:4;5:8]
A =
     1     2     3     4
     5     6     7     8

pow2(A)
ans =
     2     4     8    16
    32    64   128   256
```

2.7 数组运算和矩阵运算

2.7.1 数组运算和矩阵运算指令对照汇总

【例 2.7.1-1】两种不同转置的比较

```
clear;A=zeros(2,3);
A(:)=1:6;
A=A*(1+i)
A_A=A.'
A_M=A'
A =
   1.0000 + 1.0000i   3.0000 + 3.0000i   5.0000 + 5.0000i
   2.0000 + 2.0000i   4.0000 + 4.0000i   6.0000 + 6.0000i
A_A =
   1.0000 + 1.0000i   2.0000 + 2.0000i
   3.0000 + 3.0000i   4.0000 + 4.0000i
   5.0000 + 5.0000i   6.0000 + 6.0000i
A_M =
   1.0000 - 1.0000i   2.0000 - 2.0000i
   3.0000 - 3.0000i   4.0000 - 4.0000i
   5.0000 - 5.0000i   6.0000 - 6.0000i
```

2.8 多项式的表达方式及其操作

2.8.1 多项式的表达和创建

一 多项式表达方式的约定

二 多项式行向量的创建方法

【例 2.8.1.2-1】求 3 阶方阵 A 的特征多项式。

```
A=[11 12 13;14 15 16;17 18 19];
PA=poly(A)
PPA=poly2str(PA,'s')
PA =
    1.0000  -45.0000  -18.0000    0.0000
PPA =
   s^3 - 45 s^2 - 18 s + 1.8303e-014
```

【例 2.8.1.2-2】由给定根向量求多项式系数向量。

```
R=[-0.5,-0.3+0.4*i,-0.3-0.4*i];
```

```
P=poly(R)
PR=real(P)
PPR=poly2str(PR,'x')
P =
    1.0000    1.1000    0.5500    0.1250
PR =
    1.0000    1.1000    0.5500    0.1250
PPR =
   x^3 + 1.1 x^2 + 0.55 x + 0.125
```

2.8.2 多项式运算函数

【例 2.8.2-1】求 $\frac{(s^2+2)(s+4)(s+1)}{s^3+s+1}$ 的“商”及“余”多项式。

```
p1=conv([1,0,2],conv([1,4],[1,1]));
p2=[1 0 1 1];
[q,r]=deconv(p1,p2);
cq='商多项式为  '; cr='余多项式为  ';
disp([cq,poly2str(q,'s')]),disp([cr,poly2str(r,'s')])
商多项式为      s + 5
余多项式为      5 s^2 + 4 s + 3
```

【例 2.8.2-2】两种多项式求值指令的差别。

```
S=pascal(4)
P=poly(S);
PP=poly2str(P,'s')
PA=polyval(P,S)
PM=polyvalm(P,S)
S =
     1     1     1     1
     1     2     3     4
     1     3     6    10
     1     4    10    20
PP =
   s^4 - 29 s^3 + 72 s^2 - 29 s + 1
PA =
  1.0e+004 *
    0.0016    0.0016    0.0016    0.0016
    0.0016    0.0015   -0.0140   -0.0563
    0.0016   -0.0140   -0.2549   -1.2089
    0.0016   -0.0563   -1.2089   -4.3779
PM =
  1.0e-010 *
    0.0016    0.0033    0.0090    0.0205
    0.0045    0.0101    0.0286    0.0697
    0.0095    0.0210    0.0653    0.1596
    0.0163    0.0387    0.1226    0.3019
```

【例 2.8.2-3】部分分式展开。

```
a=[1,3,4,2,7,2];
b=[3,2,5,4,6];
[r,s,k]=residue(b,a)
r =
   1.1274 + 1.1513i
   1.1274 - 1.1513i
  -0.0232 - 0.0722i
  -0.0232 + 0.0722i
   0.7916
s =
```

```
  -1.7680 + 1.2673i
  -1.7680 - 1.2673i
   0.4176 + 1.1130i
   0.4176 - 1.1130i
  -0.2991
k =
     []
```

2.9 标准数组生成函数和数组操作函数

2.9.1 标准数组生成函数

【例 2.9.1-1】标准数组产生的演示。

```
ones(1,2)
ans =
     1     1

ones(2)
ans =
     1     1
     1     1

randn('state',0)
randn(2,3)
ans =
   -0.4326    0.1253   -1.1465
   -1.6656    0.2877    1.1909

D=eye(3)
D =
     1     0     0
     0     1     0
     0     0     1

diag(D)
ans =
     1
     1
     1

diag(diag(D))
ans =
     1     0     0
     0     1     0
     0     0     1

repmat(D,1,3)
ans =
  Columns 1 through 8
     1     0     0     1     0     0     1     0
     0     1     0     0     1     0     0     1
     0     0     1     0     0     1     0     0
  Column 9
     0
     0
     1
```

2.9.2 数组操作函数

【例 2.9.2-1】diag 与 reshape 的使用演示。

```
a=-4:4
A=reshape(a,3,3)
a =
  Columns 1 through 8
    -4    -3    -2    -1     0     1     2     3
  Column 9
     4
A =
    -4    -1     2
    -3     0     3
    -2     1     4

a1=diag(A,1)
a1 =
    -1
     3

A1=diag(a1,-1)
A1 =
     0     0     0
    -1     0     0
     0     3     0
```

【例 2.9.2-2】数组转置、对称交换和旋转操作后果的对照比较。

```
A
A =
    -4    -1     2
    -3     0     3
    -2     1     4

A.'
ans =
    -4    -3    -2
    -1     0     1
     2     3     4

flipud(A)
ans =
    -2     1     4
    -3     0     3
    -4    -1     2

fliplr(A)
ans =
     2    -1    -4
     3     0    -3
     4     1    -2

rot90(A)
ans =
     2     3     4
    -1     0     1
    -4    -3    -2
```

【例 2.9.2-3】演示 Kronecker 乘法不具备“可交换规律”。

```
B=eye(2)
C=reshape(1:4,2,2)
B =
```

```
     1     0
     0     1
C =
     1     3
     2     4

kron(B,C)
ans =
     1     3     0     0
     2     4     0     0
     0     0     1     3
     0     0     2     4

kron(C,B)
ans =
     1     0     3     0
     0     1     0     3
     2     0     4     0
     0     2     0     4
```

2.10　数组构作技法综合

【例 2.10-1】数组的扩展。

（1）数组的赋值扩展法

```
A=reshape(1:9,3,3)
A =
     1     4     7
     2     5     8
     3     6     9

A(5,5)=111
A =
     1     4     7     0     0
     2     5     8     0     0
     3     6     9     0     0
     0     0     0     0     0
     0     0     0     0   111

A(:,6)=222
A =
     1     4     7     0     0   222
     2     5     8     0     0   222
     3     6     9     0     0   222
     0     0     0     0     0   222
     0     0     0     0   111   222
```

（2）多次寻访扩展法

```
AA=A(:,[1:6,1:6])
AA =
     1     4     7     0     0   222     1     4     7     0     0   222
     2     5     8     0     0   222     2     5     8     0     0   222
     3     6     9     0     0   222     3     6     9     0     0   222
     0     0     0     0     0   222     0     0     0     0     0   222
     0     0     0     0   111   222     0     0     0     0   111   222
```

（3）合成扩展法

```
B=ones(2,6)
B =
     1     1     1     1     1     1
     1     1     1     1     1     1

AB_r=[A;B]
```

```
AB_r =
     1     4     7     0     0   222
     2     5     8     0     0   222
     3     6     9     0     0   222
     0     0     0     0     0   222
     0     0     0     0   111   222
     1     1     1     1     1     1
     1     1     1     1     1     1

AB_c=[A,B(:,1:5)']
AB_c =
     1     4     7     0     0   222     1     1
     2     5     8     0     0   222     1     1
     3     6     9     0     0   222     1     1
     0     0     0     0     0   222     1     1
     0     0     0     0   111   222     1     1
```

【例 2.10-2】提取子数组，合成新数组。

```
A
A =
     1     4     7     0     0   222
     2     5     8     0     0   222
     3     6     9     0     0   222
     0     0     0     0     0   222
     0     0     0     0   111   222

AB_BA=triu(A,1)+tril(A,-1)
AB_BA =
     0     4     7     0     0   222
     2     0     8     0     0   222
     3     6     0     0     0   222
     0     0     0     0     0   222
     0     0     0     0     0   222

AB1=[A(1:2,end:-1:1);B(1,:)]
AB1 =
   222     0     0     7     4     1
   222     0     0     8     5     2
     1     1     1     1     1     1
```

【例 2.10-3】单下标寻访和 reshape 指令演示。

```
clear
A=reshape(1:16,2,8)
A =
     1     3     5     7     9    11    13    15
     2     4     6     8    10    12    14    16

reshape(A,4,4)
ans =
     1     5     9    13
     2     6    10    14
     3     7    11    15
     4     8    12    16

s=[1 3 6 8 9 11 14 16];
A(s)=0
A =
     0     0     5     7     0     0    13    15
     2     4     0     0    10    12     0     0
```

【例 2.10-4】“对列（或行）同加一个数”三种的操作方法。

```
clear,A=reshape(1:9,3,3)
A =
```

```
     1     4     7
     2     5     8
     3     6     9

b=[1 2 3];A_b1=A-b([1 1 1],:)
A_b1 =
     0     2     4
     1     3     5
     2     4     6

A_b2=A-repmat(b,3,1)
A_b2 =
     0     2     4
     1     3     5
     2     4     6

A_b3=[A(:,1)-b(1),A(:,2)-b(2),A(:,3)-b(3)]
A_b3 =
     0     2     4
     1     3     5
     2     4     6
```

【例 2.10-5】逻辑函数的运用示例。

```
randn('state',1),R=randn(3,6)
R =
    0.8644    0.8735   -1.1027    0.1684   -0.5523   -0.6149
    0.0942   -0.4380    0.3962   -1.9654   -0.8197   -0.2546
   -0.8519   -0.4297   -0.9649   -0.7443    1.1091   -0.2698

L=abs(R)<0.5|abs(R)>1.5
L =
     0     0     0     1     0     0
     1     1     1     1     0     1
     0     1     0     0     0     1

R(L)=0
R =
    0.8644    0.8735   -1.1027         0   -0.5523   -0.6149
         0         0         0         0   -0.8197         0
   -0.8519         0   -0.9649   -0.7443    1.1091         0

s=(find(R==0))'
s =
     2     5     6     8    10    11    17    18

R(s)=111
R =
    0.8644    0.8735   -1.1027  111.0000   -0.5523   -0.6149
  111.0000  111.0000  111.0000  111.0000   -0.8197  111.0000
   -0.8519  111.0000   -0.9649   -0.7443    1.1091  111.0000

[ii,jj]=find(R==111);
disp(ii'),disp(jj')
     2     2     3     2     1     2     2     3
     1     2     2     3     4     4     6     6
```

2.11 高维数组

2.11.1　　高维数组的创建

【例 2.11.1-1】“全下标”元素赋值方式创建高维数组演示。

```
A(2,2,2)=1
A(:,:,1) =
     0     0
     0     0
A(:,:,2) =
     0     0
     0     1


B(2,5,:)=1:3
B(:,:,1) =
     0     0     0     0     0
     0     0     0     0     1
B(:,:,2) =
     0     0     0     0     0
     0     0     0     0     2
B(:,:,3) =
     0     0     0     0     0
     0     0     0     0     3
```

【例 2.11.1-2】低维数组合成高维数组。

```
clear,A=ones(2,3);A(:,:,2)=ones(2,3)*2;A(:,:,3)=ones(2,3)*3
A(:,:,1) =
     1     1     1
     1     1     1
A(:,:,2) =
     2     2     2
     2     2     2
A(:,:,3) =
     3     3     3
     3     3     3
```

【例 2.11.1-3】由函数 ones, zeros, rand, randn 直接创建标准高维数组的示例。

```
rand('state',1111),rand(2,4,3)
ans(:,:,1) =
    0.6278    0.9748    0.2585    0.6949
    0.2544    0.2305    0.0313    0.1223
ans(:,:,2) =
    0.4889    0.3898    0.8489    0.0587
    0.9138    0.3071    0.4260    0.6331
ans(:,:,3) =
    0.2802    0.2073    0.7438    0.2714
    0.4051    0.2033    0.4566    0.2421
```

【例 2.11.1-4】借助 cat, repmat, reshape 等函数构作高维数组。

（1）

```
cat(3,ones(2,3),ones(2,3)*2,ones(2,3)*3)
ans(:,:,1) =
     1     1     1
     1     1     1
ans(:,:,2) =
     2     2     2
     2     2     2
ans(:,:,3) =
     3     3     3
     3     3     3
```

（2）

```
repmat(ones(2,3),[1,1,3])
ans(:,:,1) =
     1     1     1
     1     1     1
ans(:,:,2) =
     1     1     1
     1     1     1
ans(:,:,3) =
     1     1     1
     1     1     1
```

（3）

```
reshape(1:12,2,2,3)
ans(:,:,1) =
     1     3
     2     4
ans(:,:,2) =
     5     7
     6     8
ans(:,:,3) =
     9    11
    10    12
```

2.11.2 高维数组的标识

【例 2.11.2-1】维数、大小和长度

```
clear;A=reshape(1:24,2,3,4);
dim_A=ndims(A)
size_A=size(A)
L_A=length(A)
dim_A =
     3
size_A =
     2     3     4
L_A =
     4
```

2.11.3 高维数组构作和操作函数汇总

【例 2.11.3-1】数组元素对称交换指令 flipdim 的使用示例。

```
A=reshape(1:18,2,3,3)
A(:,:,1) =
     1     3     5
     2     4     6
A(:,:,2) =
     7     9    11
     8    10    12
A(:,:,3) =
    13    15    17
    14    16    18

flipdim(A,1)
ans(:,:,1) =
     2     4     6
     1     3     5
ans(:,:,2) =
     8    10    12
     7     9    11
```

```
ans(:,:,3) =
    14    16    18
    13    15    17

flipdim(A,3)
ans(:,:,1) =
    13    15    17
    14    16    18
ans(:,:,2) =
     7     9    11
     8    10    12
ans(:,:,3) =
     1     3     5
     2     4     6
```

【例 2.11.3-2】数组的“ 维序号左移 ”重组。

```
shiftdim(A,1)
ans(:,:,1) =
     1     7    13
     3     9    15
     5    11    17
ans(:,:,2) =
     2     8    14
     4    10    16
     6    12    18

shiftdim(A,2)
ans(:,:,1) =
     1     2
     7     8
    13    14
ans(:,:,2) =
     3     4
     9    10
    15    16
ans(:,:,3) =
     5     6
    11    12
    17    18
```

【例 2.11.3-3】广义非共轭转置。

```
permute(A,[2,3,1])
ans(:,:,1) =
     1     7    13
     3     9    15
     5    11    17
ans(:,:,2) =
     2     8    14
     4    10    16
     6    12    18

permute(A,[1,3,2])
ans(:,:,1) =
     1     7    13
     2     8    14
ans(:,:,2) =
     3     9    15
     4    10    16
ans(:,:,3) =
     5    11    17
     6    12    18
```

【例 2.11.3-4】“孤维”的撤消和降维。

```
B=cat(4,A(:,:,1),A(:,:,2),A(:,:,3))
B(:,:,1,1) =
     1     3     5
     2     4     6
B(:,:,1,2) =
     7     9    11
     8    10    12
B(:,:,1,3) =
    13    15    17
    14    16    18

size(B)
ans =
     2     3     1     3

C=squeeze(B)
C(:,:,1) =
     1     3     5
     2     4     6
C(:,:,2) =
     7     9    11
     8    10    12
C(:,:,3) =
    13    15    17
    14    16    18

size(C)
ans =
     2     3     3
```

【例 2.11.3-5】赋“空阵”值操作。

```
A=reshape(1:18,2,3,3)
A(:,:,1) =
     1     3     5
     2     4     6
A(:,:,2) =
     7     9    11
     8    10    12
A(:,:,3) =
    13    15    17
    14    16    18

A(:,2:3,:)=[]
B=A;
A(:,:,1) =
     1
     2
A(:,:,2) =
     7
     8
A(:,:,3) =
    13
    14

size(A)
ans =
     2     1     3

A_1=squeeze(A)
```

```
A_1 =
     1     7    13
     2     8    14

size(B)
ans =
     2     1     3

B(:,1,:)=[]
B =
   Empty array: 2-by-0-by-3
```

2.12 "非数"和"空"数组

2.12.1 非数 NaN

【例 2.12.1-1】非数的产生和性质演示。

（1）

```
a=0/0,b=0*log(0),c=inf-inf
Warning: Divide by zero.
a =
   NaN
Warning: Log of zero.
b =
   NaN
c =
   NaN
```

（2）

```
0*a,sin(a)
ans =
   NaN
ans =
   NaN
```

（3）

```
a==nan
ans =
     0
```

（4）

```
a~=nan
a==b
b>c
ans =
     1
ans =
     0
ans =
     0
```

（5）

```
class(a)
isnan(a)
ans =
double
ans =
     1
```

【例 2.12.1-2】非数元素的寻访

```
rand('state',0)
R=rand(2,5);R(1,5)=NaN;R(2,3)=NaN
R =
    0.9501    0.6068    0.8913    0.4565       NaN
    0.2311    0.4860       NaN    0.0185    0.4447

isnan(R)
ans =
     0     0     0     0     1
     0     0     1     0     0

Linear_index=find(isnan(R))
[r_index,c_index]=ind2sub(size(R),Linear_index);
disp('r_index   c_index'),disp([r_index c_index])
Linear_index =
     6
     9
r_index   c_index
     2     3
     1     5
```

2.12.2 “ 空 ” 数组

【例 2.12.2-1】关于 “ 空 ” 数组的算例。

（1）

```
a=[],b=ones(2,0),c=zeros(2,0),d=eye(2,0),f=rand(2,3,0,4)
a =
     []
b =
   Empty matrix: 2-by-0
c =
   Empty matrix: 2-by-0
d =
   Empty matrix: 2-by-0
f =
   Empty array: 2-by-3-by-0-by-4
```

（2）

```
class(a)
isnumeric(a)
isempty(a)
ans =
double
ans =
     1
ans =
     1

which a
ndims(a)
size(a)
a is a variable.
ans =
     2
ans =
     0     0
```

（3）

```
b_c1=b.*c
b_c2=b'*c
b_c3=b*c'
b_c1 =
   Empty matrix: 2-by-0
b_c2 =
     []
b_c3 =
     0     0
     0     0
```

（4）

```
a==b
ans =
     0

b==c
ans =
   Empty matrix: 2-by-0

c>d
ans =
   Empty matrix: 2-by-0

a==0
Warning: Future versions will return empty for empty == scalar comparisons.
ans =
     0

a~=0
Warning: Future versions will return empty for empty ~= scalar comparisons.
ans =
     1
```

（5）

```
A=reshape(-4:5,2,5)
A =
    -4    -2     0     2     4
    -3    -1     1     3     5

L2=A>10
find(L2)
L2 =
     0     0     0     0     0
     0     0     0     0     0
ans =
     []
```

（6）

```
A(:,[2,4])=[]
A =
    -4     0     4
    -3     1     5
```

2.13 关系操作和逻辑操作

2.13.1 关系操作

【例 2.13.1-1】关系运算示例。

```
A=1:9,B=10-A,r0=(A<4),r1=(A==B)
A =
     1     2     3     4     5     6     7     8     9
B =
     9     8     7     6     5     4     3     2     1
r0 =
     1     1     1     0     0     0     0     0     0
r1 =
     0     0     0     0     1     0     0     0     0
```

【例 2.13.1-2】关系运算运用之一：求近似极限，修补图形缺口。

```
t=-2*pi:pi/10:2*pi;
y=sin(t)./t;
tt=t+(t==0)*eps;
yy=sin(tt)./tt;
subplot(1,2,1),plot(t,y),axis([-7,7,-0.5,1.2]),
xlabel('t'),ylabel('y'),title('残缺图形')
subplot(1,2,2),plot(tt,yy),axis([-7,7,-0.5,1.2])
xlabel('t'),ylabel('yy'),title('正确图形')
Warning: Divide by zero.
```

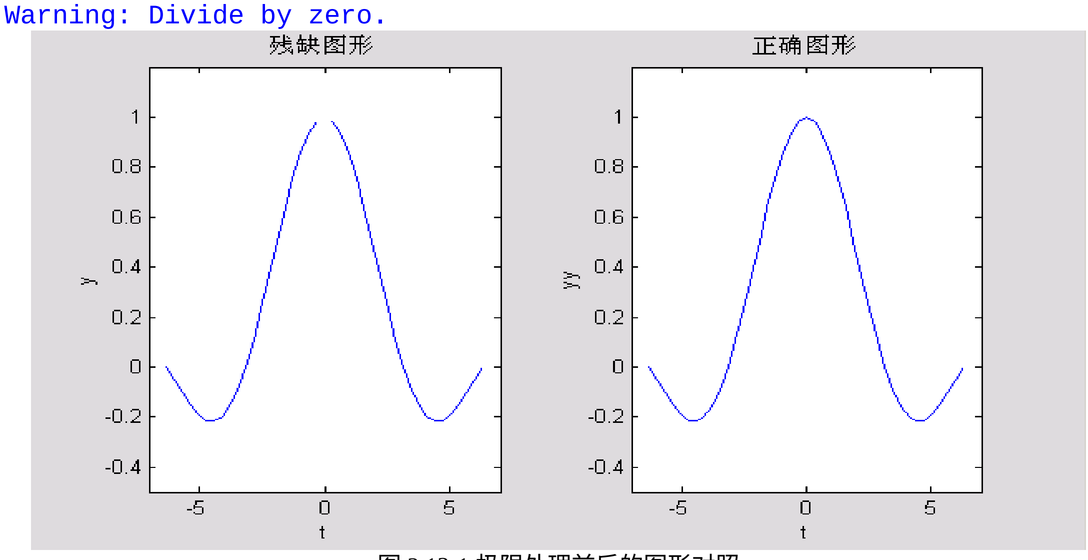


图 2.13-1 极限处理前后的图形对照

2.13.2 逻辑操作

【例 2.13.2-1】逻辑操作示例。注意逻辑运算和关系运算之间的优先级次序。（详见下节的表 2.13-5）

```
A=-3:3;
L1=~(A>0)
L2=~A>0
L3=~A
L4=A>-2&A<1
L1 =
     1     1     1     1     0     0     0
L2 =
     0     0     0     1     0     0     0
L3 =
     0     0     0     1     0     0     0
```

```
L4 =
     0     0     1     1     0     0     0
```

【例 2.13.2-2】逻辑操作应用之一：逐段解析函数的计算和表现。本例演示削顶整流正弦半波的计算和图形绘制。

```
t=linspace(0,3*pi,500);y=sin(t);
%处理方法一：
z1=((t<pi)|(t>2*pi)).*y;                       %   <3>
w=(t>pi/3&t<2*pi/3)+(t>7*pi/3&t<8*pi/3);%   <4>
w_n=~w;                                         %   <5>
z2=w*sin(pi/3)+w_n.*z1;                         %   <6>
subplot(1,3,1),plot(t,y,':r'),ylabel('y')
subplot(1,3,2),plot(t,z1,':r'),axis([0 10 -1 1])
subplot(1,3,3),plot(t,z2,'-b'),axis([0 10 -1 1])
```

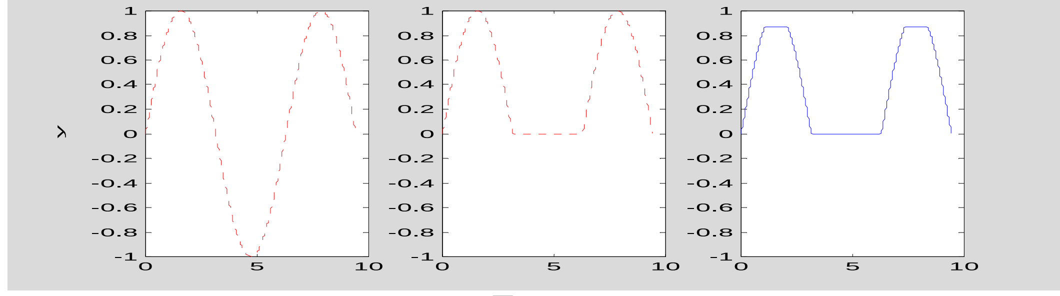


图 2.13-2

```
%处理方法二：
z=(y>=0).*y;                %   <11>
a=sin(pi/3);
z=(y>=a)*a+(y<a).*z;        %   <13>
plot(t,y,':r');hold on;plot(t,z,'-b')
xlabel('t'),ylabel('z=f(t)'),title('逐段解析函数')
legend('y=sin(t)','z=f(t)'),hold off
```

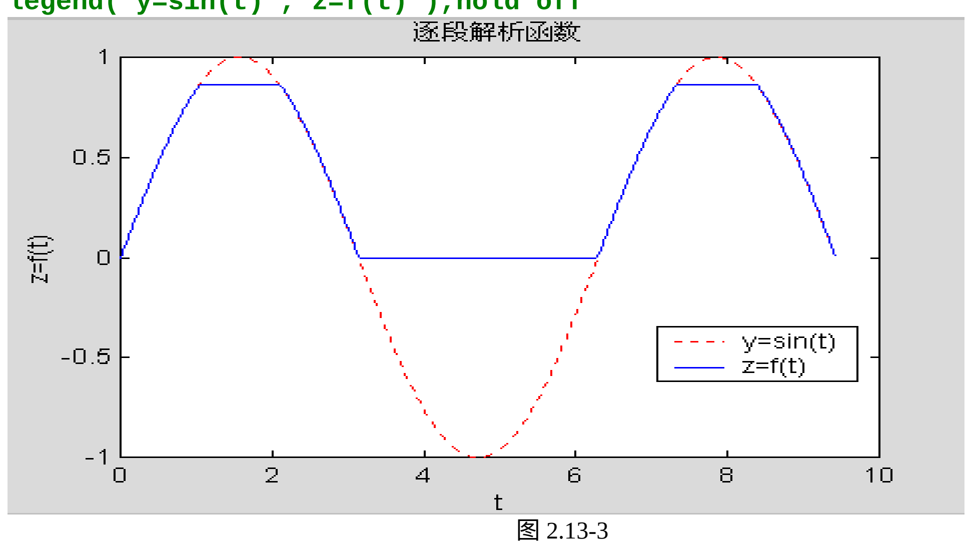


图 2.13-3

2.13.3 关系、逻辑函数

第三章 字符串、元胞和构架数组

MATLAB 6.x 版的内建数据类型（Built-in data type）就有 5 种以上，此外还有许多其他专门设计的类（Class），如符号类、内联函数类、控制工具包中的线性时不变模型类、神经网络类等。就程序设计而言，MATLAB 6.x 版采用了面向对象编程技术。数据和编程的改变使用户能更简捷而自然地解决复杂的计算问题（如符号计算问题、多变量控制系统问题、神经网络问题）。本章内容根据 MATLAB6.5 编写，但绝大部分内容适用于其他 MATLAB6.x 版本。

第二章介绍了数值数组（Numeric Array），这是读者比较熟悉的数据类型。本章将集中讲述另外三类数据：字符串数组（Character String Array）、元胞数组（Cell array）和构架数组（Structure array）。它们之间的基本差别见表 3-1。

表 3-1 四种数据类型基本构成比较表

数组类型	基本组分	组分内涵	基本组分占用字节数
数值数组	元素	双精度实数标量 或双精度复数标量	8 16
字符串数组	元素	字符	2
元胞数组	元胞	可以存放任何类型、任何大小的数据。	不定
构架数组	构架	只有挂接在构架上的“域”才能存放数据。数据可以是任何类型、任何大小。	不定

3.1 字符串数组

3.1.1 字符串入门

【例 3.1.1-1】先请读者实际操作本例，以体会数值量与字符串的区别。

```
clear
a=12345.6789
class(a)
a_s=size(a)
a =
  1.2346e+004
ans =
double
a_s =
     1     1
b='S'
class(b)
b_s=size(b)
b =
S
ans =
char
b_s =
     1     1
whos
  Name       Size              Bytes  Class

  a          1x1                   8  double array
```

```
  a_s       1x2                16  double array
  ans       1x4                 8  char array
  b         1x1                 2  char array
  b_s       1x2                16  double array

Grand total is 10 elements using 50 bytes
```

3.1.2 串数组的属性和标识

【例 3.1.2-1】本例演示：串的基本属性、标识和简单操作。

（1）

```
a='This is an example.'
a =
This is an example.
```

（2）

```
size(a)
ans =
     1    19
```

（3）

```
a14=a(1:4)
ra=a(end:-1:1)
a14 =
This
ra =
.elpmaxe na si sihT
```

（4）

```
ascii_a=double(a)
ascii_a =
  Columns 1 through 12
    84   104   105   115    32   105   115    32    97   110    32   101
  Columns 13 through 19
   120    97   109   112   108   101    46
char(ascii_a)
ans =
This is an example.
```

（5）

```
w=find(a>='a'&a<='z');
ascii_a(w)=ascii_a(w)-32;
char(ascii_a)
ans =
THIS IS AN EXAMPLE.
```

（6）

```
A='这是一个算例。';
A_s=size(A)
A56=A([5 6])
ASCII_A=double(A)
A_s =
     1     7
A56 =
算例
ASCII_A =
  Columns 1 through 6
       54754       51911       53947       47350       52195       49405
  Column 7
```

```
        41379
char(ASCII_A)
ans =
这是一个算例。
```

（7）

```
b='Example ''3.1.2-1'''
b =
Example '3.1.2-1'
```

（8）

```
ab=[a(1:7),'  ',b,' .']
ab =
This is  Example '3.1.2-1' .
```

3.1.3 复杂串数组的创建

3.1.3.1 多行串数组的直接创建

【例 3.1.3.1-1】多行串数组的直接输入示例。

```
clear
S=['This string array '
   'has multiple rows.']
S =
This string array
has multiple rows.
size(S)
ans =
     2    18
```

3.1.3.2 利用串操作函数创建多行串数组

【例 3.1.3.2-1】演示：用专门函数 **char , str2mat , strvcat** 创建多行串数组示例。

```
S1=char('This string array','has two rows.')
S1 =
This string array
has two rows.
S2=str2mat('这','字符','串数组','','由 4 行组成')
S2 =
这
字符
串数组

由 4 行组成

S3=strvcat('这','字符','串数组','','由 4 行组成')
S3 =
这
字符
串数组
由 4 行组成
size(S3)
ans =
     5     5
```

3.1.3.3 转换函数产生数码字符串

【例 3.1.3.3-1】最常用的数组/字符串转换函数 **int2str , num2str , mat2str** 示例。

（1）

```
A=eye(2,4);
A_str1=int2str(A)
A_str1 =
1  0  0  0
0  1  0  0
```

（2）

```
rand('state',0)
B=rand(2,4);
B3=num2str(B,3)
B3 =
 0.95      0.607      0.891      0.456
0.231      0.486      0.762     0.0185
```

（3）

```
B_str=mat2str(B,4)
B_str =
[0.9501 0.6068 0.8913 0.4565;0.2311 0.486 0.7621 0.0185]
Expression=['exp(-',B_str,')'];
eval(Expression)
ans =
    0.3867    0.5451    0.4101    0.6335
    0.7937    0.6151    0.4667    0.9817
```

【例 3.1.3.3-2】综合例题：在 MATLAB 计算生成的图形上标出图名和最大值点坐标。

```
clear
a=2;
w=3;
t=0:0.01:10;
y=exp(-a*t).*sin(w*t);
[y_max,i_max]=max(y);
t_text=['t=',num2str(t(i_max))];        %  <7>
y_text=['y=',num2str(y_max)];            %  <8>
max_text=char('maximum',t_text,y_text); %  <9>
%
tit=['y=exp(-',num2str(a),'t)*sin(',num2str(w),'t)'];          %<11>
plot(t,zeros(size(t)),'k')
hold on
plot(t,y,'b')
plot(t(i_max),y_max,'r.','MarkerSize',20)
text(t(i_max)+0.3,y_max+0.05,max_text)  %  <16>
title(tit),xlabel('t'),ylabel('y'),hold off
```

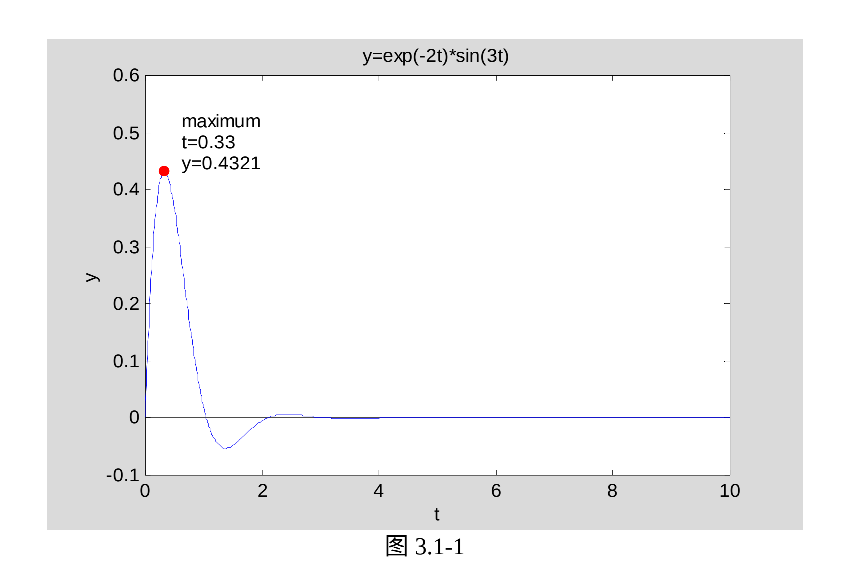


图 3.1-1

3.1.3.4 利用元胞数组创建复杂字符串

【例 3.1.3.4-1】元胞数组在存放和操作字符串上的应用。

```
a='MATLAB 6.x  ';b='includes new data types:';
c1='◆Multidimensional array';c2='◆User-definable data structure';
c3='◆Cell arrays';c4='◆Character array';
c5='◆Function handle';
c=char(c1,c2,c3,c4,c5);
C={a;b;c};                       %   <5>
disp([C{1:2}])                   %   <6>
disp(' ')                        %
disp(C{3})                       %   <8>
MATLAB 6.x  includes new data types:

◆Multidimensional array
◆User-definable data structure
◆Cell arrays
◆Character array
◆Function handle
```

3.1.4 串转换函数

【例 3.1.4-1】fprintf, sprintf, sscanf 的用法示例。

```
rand('state',0);a=rand(2,2);
s1=num2str(a)
s_s=sprintf('%.10e\n',a)
s1 =
0.95013      0.60684
0.23114      0.48598
s_s =
9.5012928515e-001
2.3113851357e-001
6.0684258354e-001
4.8598246871e-001
fprintf('%.5g\\',a)
0.95013\0.23114\0.60684\0.48598\
s_sscan=sscanf(s_s,'%f',[3,2])
s_sscan =
```

```
     0.9501    0.4860
     0.2311         0
     0.6068         0
```

3.1.5 串操作函数

3.2 元胞数组

3.2.1 元胞数组的创建和显示

3.2.1.1 元胞标识寻访和内容编址寻访的不同

3.2.1.2 元胞数组的创建和显示

【例 3.2.1.2-1】本例演示：(2×2) 元胞数组的创建。

```
C_str=char('这是','元胞数组创建算例 1');
R=reshape(1:9,3,3);
Cn=[1+2i];
S_sym=sym('sin(-3*t)*exp(-t)');
```

（1）直接创建法之一

```
A(1,1)={C_str};A(1,2)={R};A(2,1)={Cn};A(2,2)={S_sym};
A
A =
            [2x10 char]    [3x3 double]
    [1.0000+ 2.0000i]    [1x1 sym    ]
```

（2）直接创建法之二

```
B{1,1}=C_str;B{1,2}=R;B{2,1}=Cn;B{2,2}=S_sym;
celldisp(B)
B{1,1} =
这是
元胞数组创建算例 1
B{2,1} =
   1.0000 + 2.0000i
B{1,2} =
     1     4     7
     2     5     8
     3     6     9
B{2,2} =
    sin(-3*t)*exp(-t)
```

3.2.2 元胞数组的扩充、收缩和重组

【例 3.2.2-1】元胞数组的扩充。

（1）

```
C=cell(2);
C(:,1)={char('Another','text string');10:-1:1}
C =
    [2x11 char  ]     []
    [1x10 double]     []
```

（2）

```
AC=[A C]
A_C=[A;C]
AC =
          [2x10 char]    [3x3 double]    [2x11 char  ]     []
    [1.0000+ 2.0000i]    [1x1 sym   ]    [1x10 double]     []
A_C =
        [2x10 char  ]    [3x3 double]
    [1.0000+ 2.0000i]    [1x1 sym   ]
        [2x11 char  ]              []
        [1x10 double]              []
```

【例 3.2.2-2】**cellplot** 能用图形形象化地表示元胞数组的内容。（A_C 取自上例）

```
cellplot(A_C,'legend')
```

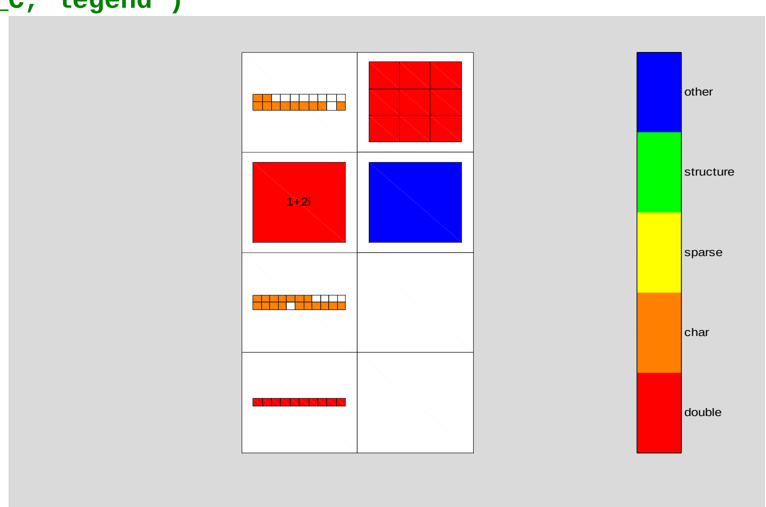


图 3.2-1

【例 3.2.2-3】元胞数组的收缩和重组。

（1）

```
A_C(3,:)=[]
A_C =
        [2x10 char  ]    [3x3 double]
    [1.0000+ 2.0000i]    [1x1 sym   ]
        [1x10 double]              []
```

（2）

```
R_A_C=reshape(A_C,2,3)
R_A_C =
          [2x10 char]    [1x10 double]    [1x1 sym]
    [1.0000+ 2.0000i]    [3x3  double]           []
```

3.2.3 元胞数组内容的调取

【例 3.2.3-1】元胞数组内容的调取示例。

（1）

```
f1=R_A_C(1,3)
class(f1)
f1 =
    [1x1 sym]
ans =
cell
```

（2）

```
f2=R_A_C{1,3}
class(f2)
f2 =
sin(-3*t)*exp(-t)
ans =
sym
```

（3）

```
f3=R_A_C{1,1}(:,[1 2 5 6])
f3 =
这是
元胞创建
```

（4）

```
[f4,f5,f6]=deal(R_A_C{[1,3,4]})
f4 =
这是
元胞数组创建算例 1
f5 =
    10     9     8     7     6     5     4     3     2     1
f6 =
     1     4     7
     2     5     8
     3     6     9
```

3.2.4 元胞数组转换函数

【例 3.2.4-1】常用元胞数组转换函示例。

（1） num2cell 把数值数组转换成元胞数组

```
rand('state',0);
A=rand(2,3,2)
C1=num2cell(A)
A(:,:,1) =
    0.9501    0.6068    0.8913
    0.2311    0.4860    0.7621
A(:,:,2) =
    0.4565    0.8214    0.6154
    0.0185    0.4447    0.7919
C1(:,:,1) =
    [0.9501]    [0.6068]    [0.8913]
    [0.2311]    [0.4860]    [0.7621]
C1(:,:,2) =
    [0.4565]    [0.8214]    [0.6154]
    [0.0185]    [0.4447]    [0.7919]

C2=num2cell(A,1)
C2(:,:,1) =
    [2x1 double]    [2x1 double]    [2x1 double]
C2(:,:,2) =
    [2x1 double]    [2x1 double]    [2x1 double]

C3=num2cell(A,[2,3])
C3 =
    [1x3x2 double]
    [1x3x2 double]
```

（2）

```
clear,x=zeros(4,5);
x(:)=1:20
C4=mat2cell(x, [2 2], [3 2])
```

```
celldisp(C4)
x =
     1     5     9    13    17
     2     6    10    14    18
     3     7    11    15    19
     4     8    12    16    20
C4 =
    [2x3 double]    [2x2 double]
    [2x3 double]    [2x2 double]
C4{1,1} =
     1     5     9
     2     6    10
C4{2,1} =
     3     7    11
     4     8    12
C4{1,2} =
    13    17
    14    18
C4{2,2} =
    15    19
    16    20
```

（3）

```
D=cell2mat(C4(1,:))
D =
     1     5     9    13    17
     2     6    10    14    18
```

3.3 构架数组

3.3.1 构架数组的创建和显示

3.3.1.1 直接创建法及显示

【例 3.3.1.1-1】本例通过温室数据（包括温室名、容积、温度、湿度等）演示：单构架的创建和显示。

（1）

```
green_house.name='一号房';               %    <1>
green_house.volume='2000 立方米';       %   <2>
green_house.parameter.temperature=[31.2    30.4    31.6    28.7
                                    29.7    31.1    30.9    29.6];%<3>
green_house.parameter.humidity=[62.1    59.5    57.7    61.5
                                 62.0    61.9    59.2    57.5];   %<4>
```

（2）显示 " 单构架 " 结构和内容

```
green_house                             %    <5>
green_house =
         name: '一号房'
       volume: '2000 立方米'
    parameter: [1x1 struct]
green_house.parameter                   %    <6>
ans =
    temperature: [2x4 double]
       humidity: [2x4 double]
green_house.parameter.temperature%    <7>
```

```
ans =
   31.2000   30.4000   31.6000   28.7000
   29.7000   31.1000   30.9000   29.6000
```

【例 3.3.1.1-2】本例演示构架数组的创建和显示，并利用构架数组保存一个温室群的数据。本例的运行以例 3.3.1.1-1 为先导。

（1）

```
green_house(2,3).name='六号房';          %   <1>
```

（2）

```
green_house                               %   <2>
green_house =
2x3 struct array with fields:
    name
    volume
    parameter
green_house(2,3)                          %   <3>
ans =
         name: '六号房'
       volume: []
    parameter: []
```

3.3.1.2 利用构造函数创建构架数组

【例 3.3.1.2-1】利用构造函数 **struct**，建立温室群的数据库。

（1）

```
a=cell(2,3);
green_house_1=struct('name',a,'volume',a,'parameter',a(1,2))   % <2>
green_house_1 =
2x3 struct array with fields:
    name
    volume
    parameter
```

（2）

```
green_house_2=struct('name',a,'volume',[],'parameter',[])      % <3>
    green_house_2 =
    2x3 struct array with fields:
        name
        volume
        parameter
```

（3）

```
green_hopuse_3(2,3)=struct('name',[],'volume',[],'parameter',[])%<4>
green_hopuse_3 =
2x3 struct array with fields:
    name
    volume
    parameter
```

（4）

```
a1={'六号房'};a2={'3200 立方米'};
green_house_4(2,3)=struct('name',a1,'volume',a2,'parameter',[]);%<6>
T6=[31.2,30.4,31.6,28.7;29.7,31.1,30.9,29.6];                  % <7>
green_house_4(2,3).parameter.temperature=T6;                   % <8>
green_house_4
green_house_4 =
```

```
2x3 struct array with fields:
    name
    volume
    parameter
```

3.3.2 构架数组域中内容的调取和设置

【例 3.3.2-1】本例目的：一，演示函数 fieldnames , getfield , setfield 的使用方法；二，让读者感受到构架数组对应用工具包的影响；三，演示 struct 函数把“ 对象 ”转换为构架的应用。本例为获得一个演练的构架，借助 Toolbox control 工具包中的 tf 函数，先产生一个用传递函数描写的 LTI 线性时不变 2 输入 2 输出系统 $\begin{bmatrix} \dfrac{3}{s^2+3s+2} & \dfrac{2}{s^2+s+1} \\ \dfrac{4s+1}{s^3+2s^2+2s+1} & \dfrac{1}{s} \end{bmatrix}$。

(1)

```
Stf=tf({3,2;[4 1],1},{[1 3 2],[1 1 1];[1 2 2 1],[1 0]})

Transfer function from input 1 to output...
             3
 #1:  -------------
      s^2 + 3 s + 2

              4 s + 1
 #2:  ---------------------
      s^3 + 2 s^2 + 2 s + 1

Transfer function from input 2 to output...
           2
 #1:  -----------
      s^2 + s + 1

      1
 #2:  -
      s
```

(2)

```
SSTF=struct(Stf)
SSTF =
         num: {2x2 cell}
         den: {2x2 cell}
    Variable: 's'
         lti: [1x1 lti]
```

(3)

```
FN=fieldnames(SSTF)
class(FN)
FN =
    'num'
    'den'
    'Variable'
    'lti'
ans =
cell
```

(4)

```
FC=getfield(SSTF,'den',{2,1})
FC{1}
```

```
poly2str(FC{1},'s')
FC =
    [1x4 double]
ans =
     1     2     2     1
ans =
   s^3 + 2 s^2 + 2 s + 1
```

（5）

```
SSTF.num{2,1}
SSTF=setfield(SSTF,'num',{2,1},{[1 3 1]});
SSTF.num{2,1}
ans =
     0     0     4     1
ans =
     1     3     1
```

3.3.3 构架数组操作深入

3.3.3.1 构架数组的扩充和收缩

【例 3.3.3.1-1】本例演示构架数组 SSTF 的扩充和收缩。（本例以例 3.3.2-1 的运行为基础。）

（1）

```
size(SSTF)
ans =
     1     1
```

（2）

```
SSTF(2,2)=struct(tf(1,[1 1]))
size(SSTF)
SSTF =
2x2 struct array with fields:
    num
    den
    Variable
    lti
ans =
     2     2
```

（3）

```
SSTF(1,:)=[]
S22n=SSTF(1,2).num,S22d=SSTF(1,2).den
printsys(S22n{1},S22d{1})
SSTF =
1x2 struct array with fields:
    num
    den
    Variable
    lti
S22n =
    [1x2 double]
S22d =
    [1x2 double]

num/den =

     1
   -----
   s + 1
```

3.3.3.2 增添域和删除域

【例 3.3.3.2-1】对构架数组 green_house 进行域的增添和删减操作。

（1）

```
clear,for k=1:10;department(k).number=['No.',int2str(k)];end
department
department =
1x10 struct array with fields:
    number
```

（2）

```
department(1).teacher=40;department(1).student=300;
department(1).PC_computer=40;
department
department =
1x10 struct array with fields:
    number
    teacher
    student
    PC_computer
```

（3）

```
department(2).teacher.male=35;department(2).teacher.female=13;
D2T=department(2).teacher     %第 2 构架 teacher 域包含两个子域
D1T=department(1).teacher     %第 1 构架 teacher 域仅是一个数
D2T =
      male: 35
    female: 13
D1T =
    40
```

（4）

```
department(2).teacher=rmfield(department(2).teacher,'male');
department(2).teacher
ans =
    female: 13
```

（5）

```
department=rmfield(department,'student')
department =
1x10 struct array with fields:
    number
    teacher
    PC_computer
department=rmfield(department,{'teacher';'PC_computer'})
department =
1x10 struct array with fields:
    number
```

3.3.3.3 数值运算操作和函数对构架数组的应用

【例 3.3.3.3-1】数值运算操作和函数在构架域上的作用。

```
n_ex=5;
for k=1:n_ex,ex(k).f=(k-1)*n_ex+[1:5];end
ex
ex =
```

```
1x5 struct array with fields:
    f

%
disp([blanks(10) '构架域中内容'])
for k=1:n_ex,disp(ex(k).f),end
          构架域中内容
     1     2     3     4     5
     6     7     8     9    10
    11    12    13    14    15
    16    17    18    19    20
    21    22    23    24    25
class(ex(1).f)
ans =
double
%
sum_f=zeros(1,5);
for k=1:n_ex,sum_f=sum_f+ex(k).f;end,sum_f
sum_f =
    55    60    65    70    75
%
disp([blanks(20) 'ex.f 的平方根值'])
for k=1:n_ex,disp(sqrt(ex(k).f)),end
                    ex.f 的平方根值
    1.0000    1.4142    1.7321    2.0000    2.2361
    2.4495    2.6458    2.8284    3.0000    3.1623
    3.3166    3.4641    3.6056    3.7417    3.8730
    4.0000    4.1231    4.2426    4.3589    4.4721
    4.5826    4.6904    4.7958    4.8990    5.0000
```

3.3.4 构架数组和元胞数组之间的转换

【例 3.3.4-1】指令 struct2cell 和 cell2struct 的使用。

(1)

```
for k=1:5,ex(k).s=['No.' int2str(k)];ex(k).f=(k-1)*5+[1:5];end
```

(2)

```
fprintf('%s\n','ex.s 域的内容   ');fprintf('%s\',blanks(4))
for k=1:5;fprintf('%s\\',[ex(k).s blanks(1)]);end
fprintf('%s\n',blanks(1)),fprintf('%s\n','ex.f 域的内容   ')
for k=1:5;disp(ex(k).f);end
ex.s 域的内容
    No.1 \No.2 \No.3 \No.4 \No.5 \
ex.f 域的内容
     1     2     3     4     5
     6     7     8     9    10
    11    12    13    14    15
    16    17    18    19    20
    21    22    23    24    25
```

(3)

```
C_ex=struct2cell(ex);
size(C_ex)
fprintf('%s\',[C_ex{1,1,1},blanks(3)])
fprintf('%5g\',C_ex{2,1,1})
ans =
     2     1     5
No.1       1    2    3    4    5
```

（4）
```
FS={'S_char';'F_num'};
EX1=cell2struct(C_ex,FS,1)
EX1 =
1x5 struct array with fields:
    S_char
    F_num
EX1(1)
ans =
    S_char: 'No.1'
    F_num: [1 2 3 4 5]
```

（5）
```
EX2=cell2struct(C_ex,'xx',2)
EX2 =
2x5 struct array with fields:
    xx
```

（6）
```
YY=strvcat('y1','y2','y3','y4','y5');EX3=cell2struct(C_ex,YY,3)
EX3 =
2x1 struct array with fields:
    y1
    y2
    y3
    y4
    y5
EX3(1)
ans =
    y1: 'No.1'
    y2: 'No.2'
    y3: 'No.3'
    y4: 'No.4'
    y5: 'No.5'
EX3(2)
ans =
    y1: [1 2 3 4 5]
    y2: [6 7 8 9 10]
    y3: [11 12 13 14 15]
    y4: [16 17 18 19 20]
    y5: [21 22 23 24 25]
```

【例 3.3.4-2】带子域的构架数组转换为元胞数组。 本例中的 ex 构架数组由例 3.3.4-1 生成，然后再运行以下程序。
```
ex(1,1).s
ans =
No.1
%
ex(1,1).s.sub='SUB 1';
ex(3,1).s.sub='SUB 3';
ex(3,1).s.num=1/3;
ex(1,1).s
ans =
    sub: 'SUB 1'
ex(3,1).s
ans =
    sub: 'SUB 3'
    num: 0.3333
C_ex_sub=struct2cell(ex)
C_ex_sub(:,:,1) =
```

```
    [1x1 struct]     []     [1x1 struct]
    [1x5 double]     []               []
C_ex_sub(:,:,2) =
    'No.2'           []      []
    [1x5 double]     []      []
C_ex_sub(:,:,3) =
    'No.3'           []      []
    [1x5 double]     []      []
C_ex_sub(:,:,4) =
    'No.4'           []      []
    [1x5 double]     []      []
C_ex_sub(:,:,5) =
    'No.5'           []      []
    [1x5 double]     []      []
size(C_ex_sub)
ans =
     2     3     5
C_ex_sub{1,1,1}
ans =
    sub: 'SUB 1'
C_ex_sub{1,3,1}
ans =
    sub: 'SUB 3'
    num: 0.3333
```

3.4 关于数据类型的归纳性说明

第四章 数值计算

4.1 引言

本章将花较大的篇幅讨论若干常见数值计算问题：线性分析、一元和多元函数分析、微积分、数据分析、以及常微分方程（初值和边值问题）求解等。但与一般数值计算教科书不同，本章的讨论重点是：如何利用现有的世界顶级数值计算资源 MATLAB。至于数学描述，本章将遵循“ 最低限度自封闭 ”的原则处理，以最简明的方式阐述理论数学、数值数学和 MATLAB 计算指令之间的内在联系及区别。

对于那些熟悉其他高级语言（如 FORTRAN，Pascal，C++）的读者来说，通过本章，MATLAB 卓越的数组处理能力、浩瀚而灵活的 M 函数指令、丰富而友善的图形显示指令将使他们体验到解题视野的豁然开朗，感受到摆脱烦琐编程后的眉眼舒展。

对于那些经过大学基本数学教程的读者来说，通过本章，MATLAB 精良完善的计算指令，自然易读的程序将使他们感悟“ 教程 ”数学的基础地位和局限性，看到从“ 理想化 ”简单算例通向科学研究和工程设计实际问题的一条途径。

对于那些熟悉 MATLAB 基本指令的读者来说，通过本章，围绕基本数值问题展开的内容将使他们体会到各别指令的运用场合和内在关系，获得综合运用不同指令解决具体问题的思路和借鉴。

由于 MATLAB 的基本运算单元是数组，所以本章内容将从矩阵分析、线性代数的数值计算开始。然后再介绍函数零点、极值的求取，数值微积分，数理统计和分析，拟合和插值，Fourier 分析，和一般常微分方程初值、边值问题。本章的最后讨论稀疏矩阵的处理，因为这只有在大型问题中，才须特别处理。

从总体上讲，本章各节之间没有依从关系，即读者没有必要从头到尾系统阅读本章内容。读者完全可以根据需要阅读有关节次。除特别说明外，每节中的例题指令是独立完整的，因此读者可以很容易地在自己机器上实践。

MATLAB 从 5.3 版升级到 6.x 版后，本章内容的变化如下：

- MATLAB 从 6.0 版起，其矩阵和特征值计算指令不再以 LINPACK 和 EISPACK 库为基础，而建筑在计算速度更快、运行更可靠的 LAPACK 和 ARPACK 程序库的新基础上。因此，虽然各种矩阵计算指令没有变化，但计算结果却可能有某些不同。这尤其突出地表现在涉及矩阵分解、特征向量、奇异向量等的计算结果上。对此，用户不必诧异，因为构成空间的基向量时不唯一的，且新版的更可信。本书新版全部算例结果是在 6.x 版上给出的。
- 在 5.3 版本中，泛函指令对被处理函数的调用是借助函数名字符串进行的。这种调用方式在 6.x 版中已被宣布为“ 过渡期内允许使用但即将被淘汰的调用方式 ”；而新的调用方式是借助“ 函数句柄 ”进行的。因此，关于述泛函指令，本章新版着重讲述如何使用“ 函数句柄 ”，同时兼顾“ 函数名字符串 ”调用法。
- MATLAB 从 6.0 版起，提供了一组专门求微分方程“ 边值问题 ”数值解的指令。适应这种变化，本章新增第 4.14.5 节，用 2 个算例阐述求解细节。
- 5.3 版中的积分指令 quad8 已经废止；6.x 版启用新积分指令 quad*l*；6.5 版新增三重积分指令 triplequad。本章新版对此作了相应的改变。

4.2 LU 分解和恰定方程组的解

4.2.1 LU 分解、行列式和逆

4.2.2 恰定方程组的解

【例 4.2.2-1】“求逆”法和“左除”法解恰定方程的性能对比

（1）

```
randn('state',0);
A=gallery('randsvd',100,2e13,2);
x=ones(100,1);
b=A*x;
cond(A)
ans =
  1.9990e+013
```

（2）

```
tic
xi=inv(A)*b;
ti=toc
eri=norm(x-xi)
rei=norm(A*xi-b)/norm(b)
ti =
    0.7700
eri =
    0.0469
rei =
    0.0047
```

（3）

```
tic;xd=A\b;
td=toc,erd=norm(x-xd),red=norm(A*xd-b)/norm(b)
td =
     0
erd =
    0.0078
red =
  2.6829e-015
```

4.2.3 范数、条件数和方程解的精度

【例 4.2.3-1】Hilbert 矩阵是著名的病态矩阵。MATLAB 中有专门的 Hilbert 矩阵及其准确逆矩阵的生成函数。本例将对方程 $Hx = b$ 近似解和准确解进行比较。

```
N=[6 8 10 12 14];
for k=1:length(N)
    n=N(k);
    H=hilb(n);
    Hi=invhilb(n);
    b=ones(n,1);
    x_approx=H\b;
    x_exact=Hi*b;
    ndb=norm(H*x_approx-b);nb=norm(b);
    ndx=norm(x_approx - x_exact);nx=norm(x_approx);
    er_actual(k)=ndx/nx;
    K=cond(H);
    er_approx(k)=K*eps;
    er_max(k)=K*ndb/nb;
end
```

```
disp('Hilbert 矩阵阶数'),disp(N)
format short e
disp('实际误差  er_actual'),disp(er_actual),disp('')
disp('近似的最大可能误差  er_approx'),disp(er_approx),disp('')
disp('最大可能误差  er_max'),disp(er_max),disp('')
Hilbert 矩阵阶数
     6     8    10    12    14
实际误差  er_actual
  1.5410e-010  1.7310e-007  1.9489e-004  9.1251e-002  2.1257e+000
近似的最大可能误差  er_approx
  3.3198e-009  3.3879e-006  3.5583e-003  3.9846e+000  9.0475e+001
最大可能误差  er_max
  7.9498e-007  3.8709e-002  1.2703e+003  4.7791e+007  4.0622e+010
```

4.3 矩阵特征值和矩阵函数

4.3.1 特征值和特征向量的求取

【例 4.3.1-1】简单实阵的特征值问题。

```
A=[1,-3;2,2/3];[V,D]=eig(A)
V =
   0.7746                0.7746
   0.0430 - 0.6310i      0.0430 + 0.6310i
D =
   0.8333 + 2.4438i           0
        0                0.8333 - 2.4438i
```

【例 4.3.1-2】本例演示：如矩阵中有元素与截断误差相当时的特性值问题。

```
A=[3          -2          -0.9       2*eps
  -2           4          -1        -eps
  -eps/4       eps/2      -1         0
  -0.5        -0.5         0.1       1   ];
[V1,D1]=eig(A);ER1=A*V1-V1*D1
[V2,D2]=eig(A,'nobalance');ER2=A*V2-V2*D2
ER1 =
    0.0000    0.0000    0.0000    0.0000
         0   -0.0000   -0.0000   -0.0000
    0.0000   -0.0000   -0.0000    0.0000
    0.0000    0.0000    0.0000   -0.5216
ER2 =
  1.0e-014 *
   -0.2665    0.0111   -0.0559   -0.1055
    0.4441    0.1221    0.0343    0.0833
    0.0022    0.0002    0.0007         0
    0.0194   -0.0222    0.0222    0.0333
```

【例 4.3.1-3】指令 **eig** 与 **eigs** 的比较。

```
rand('state',1),A=rand(100,100)-0.5;
t0=clock;[V,D]=eig(A);T_full=etime(clock,t0)
options.tol=1e-8;
options.disp=0;
t0=clock;[v,d]=eigs(A,1,'lr',options);
T_part=etime(clock,t0)
[Dmr,k]=max(real(diag(D)));
d,D(1,1)
```

```
T_full =
    0.2200
T_part =
    3.1300
d =
   3.0140 + 0.2555i
ans =
   3.0140 + 0.2555i
```

```
vk1=V(:,k);
vk1=vk1/norm(vk1);v=v/norm(v);
V_err=acos(norm(vk1'*v))*180/pi
D_err=abs(D(k,k)-d)/abs(d)
V_err =
  1.2074e-006
D_err =
  4.2324e-010
```

4.3.2 特征值问题的条件数

【例 4.3.2-1】矩阵的代数方程条件数和特征值条件数。

```
B=eye(4,4);B(3,4)=1;B
format short e,c_equ=cond(B),c_eig=condeig(B)
B =
     1     0     0     0
     0     1     0     0
     0     0     1     1
     0     0     0     1
c_equ =
  2.6180e+000
Warning: Matrix is close to singular or badly scaled.
         Results may be inaccurate. RCOND = 1.110223e-016.
> In D:\MATLAB6P1\toolbox\matlab\matfun\condeig.m at line 30
c_eig =
  1.0000e+000
  1.0000e+000
  4.5036e+015
  4.5036e+015
```

【例 4.3.2-2】对亏损矩阵进行 Jordan 分解。

```
A=gallery(5)
[VJ,DJ]=jordan(A);
[V,D,c_eig]=condeig(A);c_equ=cond(A);
DJ,D,c_eig,c_equ
A =
          -9          11         -21          63        -252
          70         -69         141        -421        1684
        -575         575       -1149        3451      -13801
        3891       -3891        7782      -23345       93365
        1024       -1024        2048       -6144       24572
DJ =
     0     1     0     0     0
     0     0     1     0     0
     0     0     0     1     0
     0     0     0     0     1
     0     0     0     0     0
D =
  Columns 1 through 4
  -0.0408                  0                  0                  0
        0           -0.0119 + 0.0386i         0                  0
```

```
        0                    0                   -0.0119 - 0.0386i          0
        0                  0                   0             0.0323 + 0.0230i
        0                    0                       0                    0
  Column 5
        0
        0
        0
        0
   0.0323 - 0.0230i
c_eig =
  1.0e+010 *
    2.1293
    2.0796
    2.0796
    2.0020
    2.0020
c_equ =
  2.0253e+018
```

4.3.3 复数特征值对角阵与实数块特征值对角阵的转化

【例 4.3.3-1】把例 4.3.1-1 中的复数特征值对角阵 D 转换成实数块对角阵，使 VR*DR/VR=A。

```
A=[1,-3;2,2/3];[V,D]=eig(A);
[VR,DR]=cdf2rdf(V,D)
VR =
    0.7746         0
    0.0430   -0.6310
DR =
    0.8333    2.4438
   -2.4438    0.8333
```

4.3.4 矩阵的谱分解和矩阵函数

【例 4.3.4-1】数组乘方与矩阵乘方的比较。

```
clear,A=[1 2 3;4 5 6;7 8 9];
A_Ap=A.^0.3
A_Mp=A^0.3
A_Ap =
    1.0000    1.2311    1.3904
    1.5157    1.6207    1.7118
    1.7928    1.8661    1.9332
A_Mp =
   0.6962 + 0.6032i   0.4358 + 0.1636i   0.1755 - 0.2759i
   0.6325 + 0.0666i   0.7309 + 0.0181i   0.8292 - 0.0305i
   0.5688 - 0.4700i   1.0259 - 0.1275i   1.4830 + 0.2150i
```

【例 4.3.4- 2】标量的数组乘方和矩阵乘方的比较。（A 取自例 4.3.4-1）

```
pA_A=(0.3).^A
pA_M=(0.3)^A
pA_A =
    0.3000    0.0900    0.0270
    0.0081    0.0024    0.0007
    0.0002    0.0001    0.0000
pA_M =
    2.9342    0.4175   -1.0993
   -0.0278    0.7495   -0.4731
   -1.9898   -0.9184    1.1531
```

【例 4.3.4-3】sin 的数组运算和矩阵运算比较。（A 取自例 4.3.4-1）

```
A_sinA=sin(A)
A_sinM=funm(A,'sin')
A_sinA =
    0.8415    0.9093    0.1411
   -0.7568   -0.9589   -0.2794
    0.6570    0.9894    0.4121
A_sinM =
   -0.6928   -0.2306    0.2316
   -0.1724   -0.1434   -0.1143
    0.3479   -0.0561   -0.4602
```

4.4 奇异值分解

4.4.1 奇异值分解和矩阵结构

4.4.1.1 奇异值分解定义

4.4.1.2 矩阵结构的奇异值分解描述

4.4.2 线性二乘问题的解

4.4.2.1 矩阵除运算的广义化

4.4.2.2 线性模型的最小二乘解

【例 4.4.2.2-1】对于超定方程 $y = Ax$，进行三种解法比较。其中 A 取 MATLAB 库中的特殊函数生成。

（1）

```
A=gallery(5);A(:,1)=[];y=[1.7 7.5 6.3 0.83 -0.082]';
x=inv(A'*A)*A'*y,xx=pinv(A)*y,xxx=A\y
Warning: Matrix is close to singular or badly scaled.
         Results may be inaccurate. RCOND = 5.405078e-018.
x =
  3.4811e+000
  5.1595e+000
  9.5340e-001
 -4.6569e-002
xx =
  3.4759e+000
  5.1948e+000
  7.1207e-001
 -1.1007e-001
Warning: Rank deficient, rank = 3  tol =   1.0829e-010.
xxx =
  3.4605e+000
  5.2987e+000
            0
 -2.9742e-001
```

（2）

```
nx=norm(x),nxx=norm(xx),nxxx=norm(xxx)
```

```
nx =
  6.2968e+000
nxx =
  6.2918e+000
nxxx =
  6.3356e+000
```

（3）

```
e=norm(y-A*x),ee=norm(y-A*xx),eee=norm(y-A*xxx)
e =
  6.9863e-001
ee =
  4.7424e-002
eee =
  4.7424e-002
```

4.5 函数的数值导数和切平面

4.5.1 法线

【例 4.5.1-1】曲面法线演示。

```
y=-1:0.1:1;x=2*cos(asin(y));
[X,Y,Z]=cylinder(x,20);
surfnorm(X(:,11:21),Y(:,11:21),Z(:,11:21));
view([120,18])
```

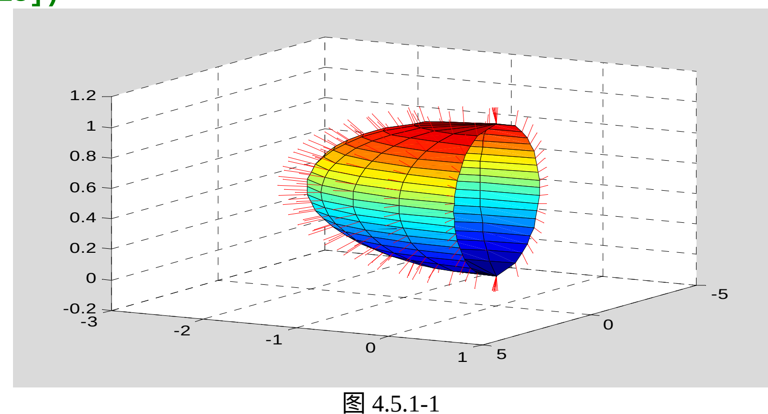


图 4.5.1-1

4.5.2 偏导数和梯度

4.5.2.1 理论定义

4.5.2.2 数值计算指令

【例 4.5.2.2-1】用一个简单矩阵表现 diff 和 gradient 指令计算方式。

```
F=[1,2,3;4,5,6;7,8,9]
Dx=diff(F)
Dx_2=diff(F,1,2)
[FX,FY]=gradient(F)
[FX_2,FY_2]=gradient(F,0.5)
F =
     1     2     3
     4     5     6
     7     8     9
```

```
Dx =
     3     3     3
     3     3     3
Dx_2 =
     1     1
     1     1
     1     1
FX =
     1     1     1
     1     1     1
     1     1     1
FY =
     3     3     3
     3     3     3
     3     3     3
FX_2 =
     2     2     2
     2     2     2
     2     2     2
FY_2 =
     6     6     6
     6     6     6
     6     6     6
```

【例 4.5.2.2-2】研究偶极子(Dipole)的电势（Electric potential）和电场强度（Electric field density）。设在 (a,b) 处有电荷 $+q$，在 $(-a,-b)$ 处有电荷 $-q$。那么在电荷所在平面上任何一点的电势和场强分别为 $V(x,y)=\frac{q}{4\pi\varepsilon_0}(\frac{1}{r_+}-\frac{1}{r_-})$，$\vec{E}=-\nabla V$。其中 $r_+=\sqrt{(x-a)^2+(y-b)^2}, r_-=\sqrt{(x+a)^2+(y+b)^2}$。$\frac{1}{4\pi\varepsilon_0}=9\cdot 10^9$。又设电荷 $q=2\cdot 10^{-6}$，$a=1.5$，$b=-1.5$。

```
clear;clf;q=2e-6;k=9e9;a=1.5;b=-1.5;x=-6:0.6:6;y=x;
[X,Y]=meshgrid(x,y);
rp=sqrt((X-a).^2+(Y-b).^2);rm=sqrt((X+a).^2+(Y+b).^2);
V=q*k*(1./rp-1./rm);
[Ex,Ey]=gradient(-V);
AE=sqrt(Ex.^2+Ey.^2);Ex=Ex./AE;Ey=Ey./AE;
cv=linspace(min(min(V)),max(max(V)),49);
contourf(X,Y,V,cv,'k-')
%axis('square')
title('\fontname{隶书}\fontsize{22}偶极子的场'),hold on
quiver(X,Y,Ex,Ey,0.7)
plot(a,b,'wo',a,b,'w+')
plot(-a,-b,'wo',-a,-b,'w-')
xlabel('x');ylabel('y'),hold off
```

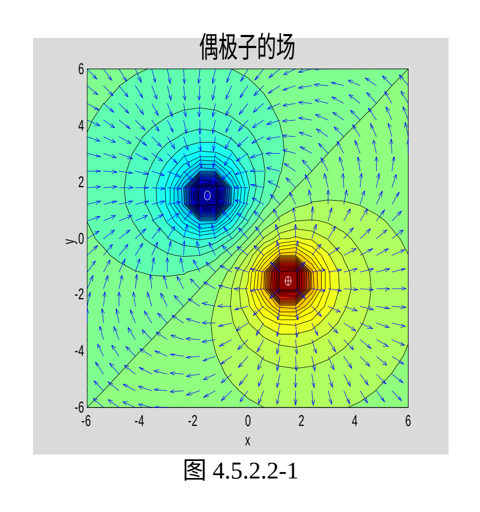


图 4.5.2.2-1

4.6 函数的零点

4.6.1 多项式的根

4.6.2 一元函数的零点

4.6.2.1 利用 MATLAB 作图指令获取初步近似解

4.6.2.2 任意一元函数零点的精确解

【例 4.6.2.2-1】通过求 $f(t)=(\sin^2 t)e^{-at}-b|t|$ 的零点，综合叙述相关指令的用法。

（1）

```
y=inline('sin(t)^2*exp(-a*t)-b*abs(t)','t','a','b');            %<1>
```

（2）

```
a=0.1;b=0.5;t=-10:0.01:10;
y_char=vectorize(y);       %    <3>
Y=feval(y_char,t,a,b);
clf,plot(t,Y,'r');hold on,plot(t,zeros(size(t)),'k');
xlabel('t');ylabel('y(t)'),hold off
```

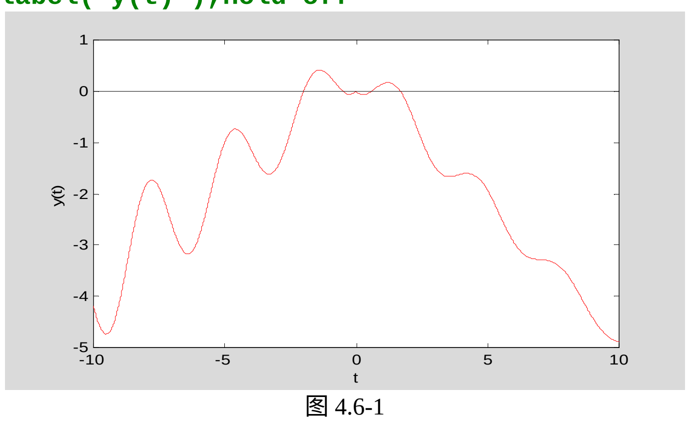


图 4.6-1

（3）

由于 Notebook 中无法实现 zoom、ginput 指令涉及的图形和鼠标交互操作，因此下面指令必须在 MATLAB 指令窗中运行，并得到如图 4.6-2 所示的局部放大图及鼠标操作线。

```
zoom on
```

```
[tt,yy]=ginput(5);zoom off
```

图 4.6-2

```
tt
tt =
   -2.0032
   -0.5415
   -0.0072
    0.5876
    1.6561
```

（4）

```
[t4,y4,exitflag]=fzero(y,tt(4),[],a,b)                              %<11>
t4 =
    0.5993
y4 =
     0
exitflag =
     1
```

（5）

```
[t3,y3,exitflag]=fzero(y,tt(3),[],a,b)
t3 =
   -0.5198
y3 =
 -5.5511e-017
exitflag =
     1
```

（6）

```
op=optimset('fzero')
op =
       ActiveConstrTol: []
               ......
               Display: 'notify'
               ......
                  TolX: 2.2204e-016
              TypicalX: []

op=optimset('tolx',0.01);
op.TolX
ans =
0.0100
```

（7）

```
[t4n,y4n,exitflag]=fzero(y,tt(4),op,a,b)
t4n =
    0.6042
y4n =
```

```
    0.0017
exitflag =
     1
```

4.6.3 多元函数的零点

【例 4.6.3-1】求解二元函数方程组 $\begin{cases} f_1(x,y)=\sin(x-y)=0 \\ f_2(x,y)=\cos(x+y)=0 \end{cases}$ 的零点。

（1）

```
x=-2:0.5:2;y=x;[X,Y]=meshgrid(x,y);
F1=sin(X-Y);F2=cos(X+Y);
v=[-0.2, 0, 0.2];
contour(X,Y,F1,v)
hold on,contour(X,Y,F2,v),hold off
```

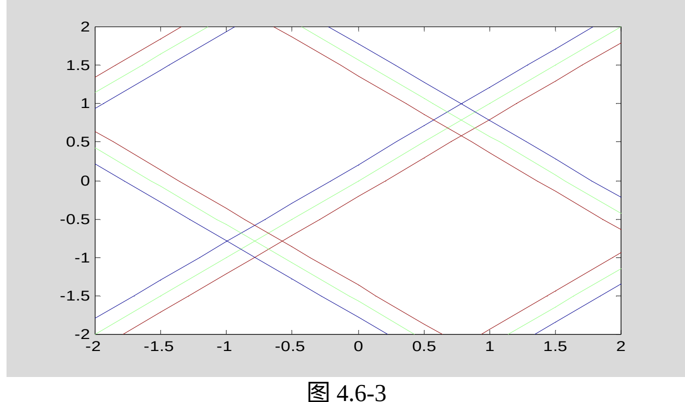


图 4.6-3

（2）

```
[x0,y0]=ginput(2);
disp([x0,y0])
   -0.7926   -0.7843
    0.7926    0.7843
```

（3）

```
fun='[sin(x(1)-x(2)),cos(x(1)+x(2))]';                        %<12>
[xy,f,exit]=fsolve(fun,[x0(2),y0(2)])                          %<13>
Optimization terminated successfully:
 First-order optimality less than OPTIONS.TolFun, and no negative/zero
curvature detected
xy =
    0.7854    0.7854
f =
  1.0e-006 *
   -0.0984    0.2011
exit =
     1
```

〖说明〗

[fun.m]

```
function ff=fun(x)
ff(1)=sin(x(1)-x(2));
ff(2)=cos(x(1)+x(2));
```

4.7 函数极值点

4.7.1 一元函数的极小值点

4.7.2 多元函数的极小值点

【例 4.7.2-1】求 $f(x,y)=100(y-x^2)^2+(1-x)^2$ 的极小值点。它即是著名的 Rosenbrock's "Banana" 测试函数。该测试函数有一片浅谷，许多算法难以越过此谷。（演示本例搜索过程的文件名为 exm04072_1_1.m 。）

（1）

```
ff=inline('100*(x(2)-x(1)^2)^2+(1-x(1))^2','x');
```

（2）

```
x0=[-1.2,1];[sx,sfval,sexit,soutput]=fminsearch(ff,x0)
sx =
    1.0000    1.0000
sfval =
  8.1777e-010
sexit =
     1
soutput =
    iterations: 85
     funcCount: 159
     algorithm: 'Nelder-Mead simplex direct search'
```

（3）

```
[ux,sfval,uexit,uoutput,grid,hess]=fminunc(ff,x0)
Warning: Gradient must be provided for trust-region method;
   using line-search method instead.

> In D:\MATLAB6P1\toolbox\optim\fminunc.m at line 211

Optimization terminated successfully:
 Current search direction is a descent direction, and magnitude of
 directional derivative in search direction less than 2*options.TolFun
ux =
    1.0000    1.0000
sfval =
  1.9116e-011
uexit =
     1
uoutput =
       iterations: 26
        funcCount: 162
         stepsize: 1.2992
    firstorderopt: 5.0020e-004
        algorithm: 'medium-scale: Quasi-Newton line search'
grid =
  1.0e-003 *
   -0.5002
   -0.1888
hess =
  820.4028 -409.5496
 -409.5496  204.7720
```

4.8 数值积分

4.8.1 一元函数的数值积分

4.8.1.1 闭型数值积分

【例 4.8.1.1-1】求 $I=\int_0^1 e^{-x^2}dx$ ，其精确值为 0.74684204Λ 。

（1）

```
syms x;IS=int('exp(-x*x)','x',0,1)
vpa(IS)
IS =
1/2*erf(1)*pi^(1/2)
ans =
.74682413281242702539946743613185
```

（2）

```
fun=inline('exp(-x.*x)','x');
Isim=quad(fun,0,1),IL=quadl(fun,0,1)
Isim =
    0.7468
IL =
    0.7468
```

（3）

```
Ig=gauss10(fun,0,1)
Ig =
    0.7463
```

（4）

```
xx=0:0.1:1.5;ff=exp(-xx.^2);
pp=spline(xx,ff);
int_pp=fnint(pp);
Ssp=ppval(int_pp,[0,1])*[-1;1]
Ssp =
    0.7468
```

（5）

图 4.8-1

4.8.1.2 开型数值积分

[gauss10.m]

```
function g = gauss10(fun,a,b)
%GAUSS10(fun,a,b)
%         fun
%=====================================================
x = [0.1488743390;0.4333953941;0.6974095683;...
     0.8650633667;0.9739065285];
w = [0.2955242247;0.2692667193;0.2190863625;...
     0.1494513492;0.0666713443];
t = .5*(b+a)+.5*(b-a)*[-flipud(x);x];
W = [flipud(w);w];
```

```
g = sum(W.*feval(fun,t))*(b-a)/2;
```

【例 4.8.1.2-1】当 $f(x)=\cos(x)$ 时，比较解析积分和近似积分。

（1）

```
syms x;F=int('cos(x)','x',-1,1),vpa(F)
F =
2*sin(1)
ans =
1.6829419696157930133050046432606
```

（2）

```
aF=cos(1/sqrt(3))+cos(-1/sqrt(3))
aF =
    1.6758
```

【例 4.8.1.2-2】求 $I=\int_0^1\sqrt{\ln\frac{1}{x}}dx$ ，准确结果是 $\frac{\sqrt{\pi}}{2}=.88622692\Lambda$ 。

（1）

```
syms x;IS=vpa(int('sqrt(log(1/x))','x',0,1))
Warning: Explicit integral could not be found.
> In D:\MATLAB6P5\toolbox\symbolic\@sym\int.m at line 58
  In D:\MATLAB6P5\toolbox\symbolic\@char\int.m at line 9
IS =
.88622692545275801364908374167057
```

（2）用 quad 指令求积

```
ff=inline('sqrt(log(1./x))','x');Isim=quad(ff,0,1)
Warning: Divide by zero.
> In D:\MATLAB6P5\toolbox\matlab\funfun\inlineeval.m at line 13
  In D:\MATLAB6P5\toolbox\matlab\funfun\@inline\feval.m at line 34
  In D:\MATLAB6P5\toolbox\matlab\funfun\quad.m at line 62
Isim =
    0.8862
```

（3）

```
Ig=gauss10(ff,0,1)
Ig =
    0.8861
```

4.8.2 多重数值积分

4.8.2.1 积分限为常数的二重积分指令

【例 4.8.2.1-1】计算 $S_{x01}=\int_1^2\left[\int_0^1 x^y dx\right]dy$ 和 $S_{x12}=\int_0^1\left[\int_1^2 x^y dx\right]dy$ 。

（1）

```
syms x y
ssx01=vpa(int(int(x^y,x,0,1),y,1,2))
ssx12=vpa(int(int(x^y,y,0,1),x,1,2))
Warning: Explicit integral could not be found.
> In D:\MATLAB6P5\toolbox\symbolic\@sym\int.m at line 58
ssx01 =
.40546510810816438197801311546435
ssx12 =
1.2292741343616127837489278679215
```

（2）

```
zz=inline('x.^y','x','y');
nsx01=dblquad(zz,0,1,1,2)
nsx12=dblquad(zz,1,2,0,1)
nsx01 =
    0.4055
nsx12 =
    1.2293
```

4.8.2.2 内积分限为函数的二重积分

[double_int.m]

```
function SS=double_int(fun,innlow,innhi,outlow,outhi)
%double_int
%
%fun
%innlow,innhi
%outlow,outhi
y1=outlow;y2=outhi;x1=innlow;x2=innhi;f_p=fun;
SS=quad(@G_yi,y1,y2,[],[],x1,x2,f_p);
```

[G_yi.m]

```
function f=G_yi(y,x1,x2,f_p)
%G_yi
%y
%x1,x2
%
%f_p
y=y(:);n=length(y);
if isnumeric(x1)==1;xx1=x1*ones(size(y));else xx1=feval(x1,y);end
if isnumeric(x2)==1;xx2=x2*ones(size(y));else xx2=feval(x2,y);end
for i=1:n
    f(i)=quad(f_p,xx1(i),xx2(i),[],[],y(i));
end
f=f(:);
```

【例 4.8.2.2-1】计算 $I = \int_1^4 \left[\int_{\sqrt{y}}^2 (x^2 + y^2) dx \right] dy$ 。

（1）

[x_low.m]

```
function f=x_low(y)
f=sqrt(y);
```

（2）

（3）

```
ff=inline('x.^2+y.^2','x','y');
SS=double_int(ff,@x_low,2,1,4)
Warning: Minimum step size reached; singularity possible.
> In D:\MATLAB6P5\toolbox\matlab\funfun\quad.m at line 88
  In D:\MATLAB6p5\work\G_yi.m at line 11
  In D:\MATLAB6P5\toolbox\matlab\funfun\@inline\feval.m at line 20
  In D:\MATLAB6P5\toolbox\matlab\funfun\quad.m at line 62
```

```
  In D:\MATLAB6p5\work\double_int.m at line 8
SS =
    9.5810
```

（4）

```
Ssym=vpa(int(int('x^2+y^2','x','sqrt(y)',2),'y',1,4))
Ssym =
9.5809523809523809523809523809524
```

4.8.3 卷积

4.8.3.1 “完整”离散序列的数值卷积

4.8.3.2 “截尾”离散序列的数值卷积

4.8.3.3 多项式乘法与离散卷积的算法同构

4.8.3.4 连续时间函数的数值卷积

4.8.3.5 卷积的 MATLAB 实现

【例 4.8.3.5-1】有序列 $A(n)=\begin{cases}1 & n=3,4,\Lambda\ ,12\\ 0 & else\end{cases}$ 和 $B(n)=\begin{cases}1 & n=2,3,\Lambda\ ,9\\ 0 & else\end{cases}$。

（A）

```
%
a=ones(1,10);n1=3;n2=12;
b=ones(1,8);n3=2;n4=9;
c=conv(a,b);nc1=n1+n3;nc2=n2+n4;
kc=nc1:nc2;
%
aa=a(1:6);nn1=3;nn2=8;
cc=conv(aa,b);ncc1=nn1+n3;
nx=nn2+n4;
ncc2=min(nn1+n4,nn2+n3);
kx=(ncc2+1):nx;kcc=ncc1:ncc2;N=length(kcc);
stem(kcc,cc(1:N),'r','filled')
axis([nc1-2,nc2+2,0,10]),grid,hold on
stem(kc,c,'b'),stem(kx,cc(N+1:end),'g'),hold off
```

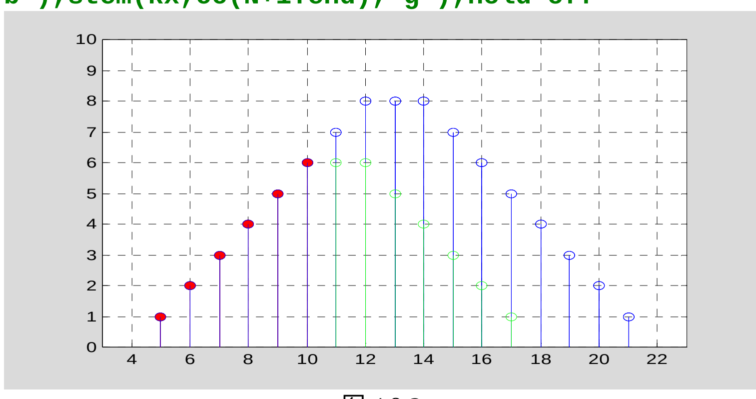


图 4.8-2

【例 4.8.3.5-2】求函数 $u(t)=e^{-t}U(t)$ 和 $h(t)=te^{-t/2}U(t)$ 的卷积。本例展示：（A）符号 Laplace 变换求卷积的理论表示；（B）SIMULINK 卷积法的执行过程和它的快速精确性。（C）从理论符号解产生相应的理论数值序列。

（1）

```
syms tao;t=sym('t','positive');
US1=laplace(exp(-t));
HS1=laplace(t*exp(-t/2))
yt1=simple(ilaplace(US1*HS1))
HS1 =
1/(1/2+s)^2
yt1 =
4*exp(-t)+(2*t-4)*exp(-1/2*t)
```

（2）

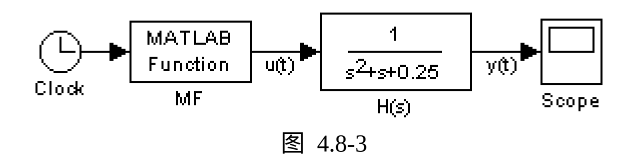


图 4.8-3

（3）

```
t=yt2(:,1);
yyt1=eval(vectorize(char(yt1)));
[dy,kd]=max(abs(yyt1-yt2(:,2)));
dy12=dy/abs(yyt1(kd))
dy12 =
  2.8978e-006
```

【例 4.8.3.5-3】用“零阶”近似法求 $u(t)=e^{-t}U(t)$ 和 $h(t)=te^{-t/2}U(t)$ 的卷积。本例演示：（A）连续函数的有限长度采样。（B）卷积数值计算三个误差（“截尾”误差、“零阶”近似误差、计算机字长误差）的影响。（C）卷积“无截尾误差”区间、“非平凡”区间端点的确定。（D）离散卷积和连续卷积之间的关系。（E）指令 **conv** 的使用。（F）绘图分格线的运用。

（1）

（2）

```
%
t2=3;t4=11;T=0.01;
tu=0:T:t2;N2=length(tu);
th=0:T:t4;N4=length(th);
u=exp(-tu);h=th.*exp(-th/2);
tx=0:T:(t2+t4);Nx=length(tx);
yt3=T*conv(u,h);
%
t=tx;yyt1=eval(vectorize(char(yt1)));
[dy,kd]=max(abs(yyt1(1:N2)-yt3(1:N2)));
dy13(1)=dy/abs(yyt1(kd));
[dy,kd]=max(abs(yyt1(N2+1:N4)-yt3(N2+1:N4)));
dy13(2)=dy/abs(yyt1(N2+kd));
[dy,kd]=max(abs(yyt1(N4+1:Nx)-yt3(N4+1:Nx)));
dy13(3)=dy/abs(yyt1(N4+kd));
```

（3）

```
disp('与理论结果的相对误差')
disp([blanks(4),'[0,3]段    [3,11]段   [11,14]段']),disp(dy13)
plot(t,yyt1,':b',tx,yt3,'r')
```

```
set(gca,'Xtick',[0,3,11,14]),grid
与理论结果的相对误差
    [0,3]段    [3,11]段   [11,14]段
    0.0068     0.0810     0.6974
```

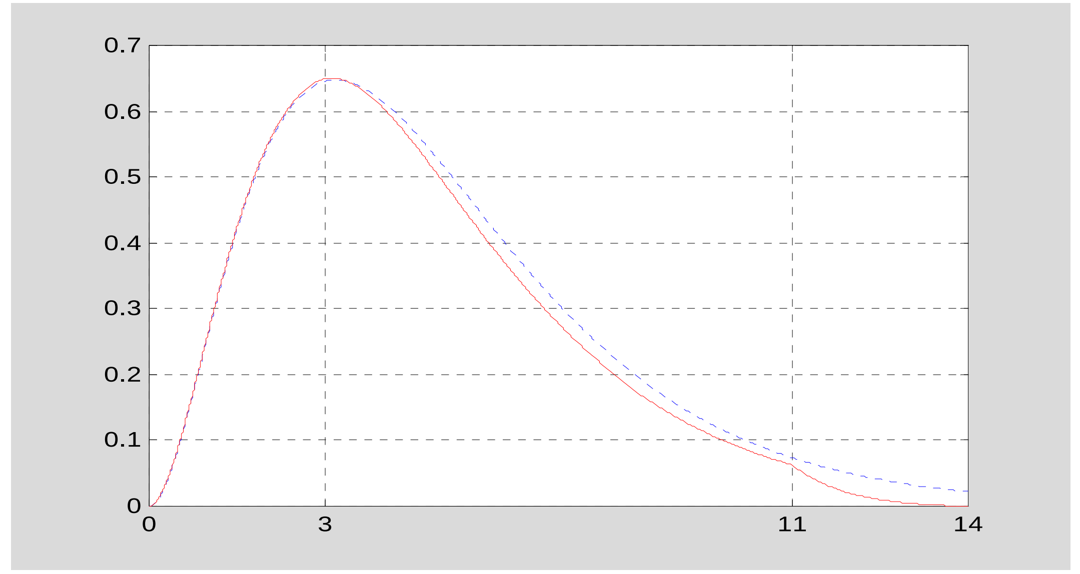


图 4.8.3.5-3-1

4.9 随机数据的统计描述

4.9.1 统计分布的数字特征

【例 4.9.1-1】样本统计特征量计算示例。

```
%
X(:,1)=ones(10,1);X(1,1)=100;X(10,1)=0.01;
rand('state',1);randn('state',1)
X(:,2)=rand(10,1);
X(:,3)=randn(10,1);X(:,3)=2*abs(min(X(:,3)))+X(:,3);
%
Moment1.arithmetic=mean(X);Moment1.median=median(X);
Moment1.geometric=geomean(X);Moment1.harmmonic=harmmean(X);
%
Moment2.Standard=std(X);Moment2.variance=var(X);
Moment2.absolute=mad(X);Moment2.range=range(X);
%
X,Moment1,Moment2
X =
  100.0000    0.9528    3.0699
    1.0000    0.7041    2.2997
    1.0000    0.9539    1.3535
    1.0000    0.5982    3.0790
    1.0000    0.8407    1.7674
    1.0000    0.4428    1.7758
    1.0000    0.8368    1.1027
    1.0000    0.5187    2.6017
    1.0000    0.0222    1.2405
    0.0100    0.3759    2.3739
Moment1 =
    arithmetic: [10.8010 0.6246 2.0664]
        median: [1 0.6511 2.0377]
```

```
      geometric: [1 0.4691 1.9463]
      harmmonic: [0.0926 0.1682 1.8276]
Moment2 =
     Standard: [31.3429 0.2951 0.7273]
     variance: [982.3760 0.0871 0.5289]
     absolute: [17.8398 0.2331 0.6184]
        range: [99.9900 0.9317 1.9762]
```

4.9.2 样本分布的频数直方图描述

【例 4.9.2-1】**hist** 指令的使用示例。

```
randn('state',1),rand('state',31)
x=randn(100,1);y=rand(100,1);
%
subplot(1,2,1),hist(x,7)
subplot(1,2,2),histfit(x,20)
```

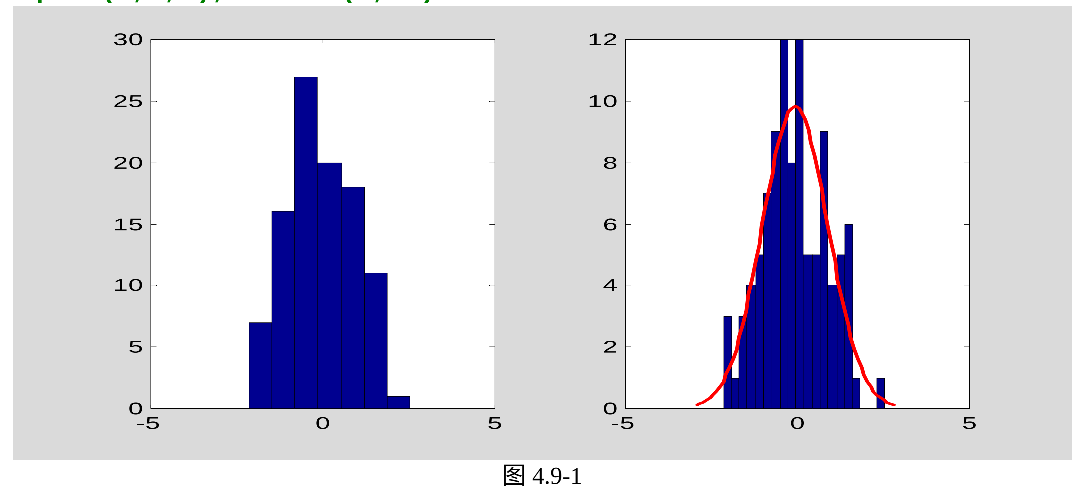


图 4.9-1

```
%
n_y1=min(y):0.1:max(y);n_y2=min(y):0.05:max(y);
subplot(1,2,1),hist(y,n_y1)
subplot(1,2,2),hist(y,n_y2)
```

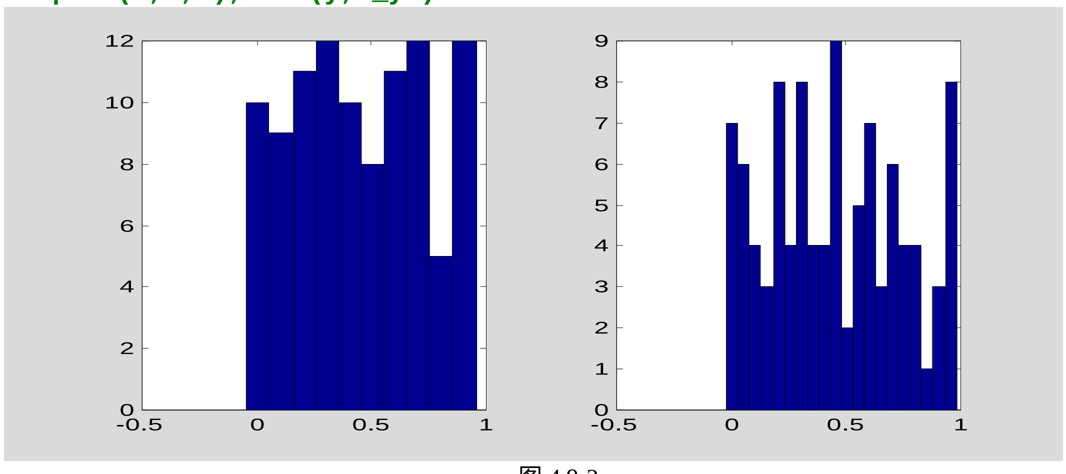


图 4.9-2

4.9.3 概率函数、分布函数、逆分布函数和随机数的发生

4.9.3.1 泊松分布(Poisson distribution)

【例 4.9.3.1-1】 泊松分布与正态分布的关系

（1）

```
Lambda=20;x=0:50;yd_p=poisspdf(x,Lambda);
yd_n=normpdf(x,Lambda,sqrt(Lambda));
```

（2）

```
plot(x,yd_n,'b-',x,yd_p,'r+')
text(30,0.07,'\fontsize{12} {\mu} = {\lambda} = 20')
```

图 4.9-3

4.9.3.2 正态分布（Normal distribution）

【例 4.9.3.2-1】正态分布标准差意义的图示。

```
mu=3;sigma=0.5;
x=mu+sigma*[-3:-1,1:3];yf=normcdf(x,mu,sigma);
P=[yf(4)-yf(3),yf(5)-yf(2),yf(6)-yf(1)];
xd=1:0.1:5;yd=normpdf(xd,mu,sigma);
%
for k=1:3
    xx{k}=x(4-k):sigma/10:x(3+k);
    yy{k}=normpdf(xx{k},mu,sigma);
end
subplot(1,3,1),plot(xd,yd,'b');hold on
fill([x(3),xx{1},x(4)],[0,yy{1},0],'g')
text(mu-0.5*sigma,0.3,num2str(P(1))),hold off
subplot(1,3,2),plot(xd,yd,'b');hold on
fill([x(2),xx{2},x(5)],[0,yy{2},0],'g')
text(mu-0.5*sigma,0.3,num2str(P(2))),hold off
subplot(1,3,3),plot(xd,yd,'b');hold on
fill([x(1),xx{3},x(6)],[0,yy{3},0],'g')
text(mu-0.5*sigma,0.3,num2str(P(3))),hold off
```

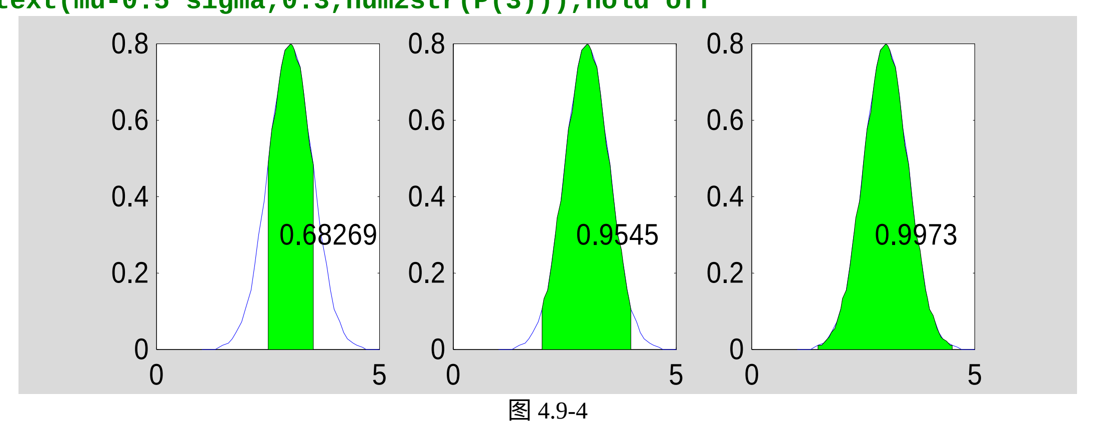


图 4.9-4

4.9.3.3 χ^2 分布（Chi-square distribution）

【例 4.9.3.3-1】 演示逆累计分布函数的应用。

```
v=4;xi=0.9;x_xi=chi2inv(xi,v);
x=0:0.1:15;yd_c=chi2pdf(x,v);
%。
plot(x,yd_c,'b'),hold on
xxf=0:0.1:x_xi;yyf=chi2pdf(xxf,v);
fill([xxf,x_xi],[yyf,0],'g')
text(x_xi*1.01,0.01,num2str(x_xi))
text(10,0.16,['\fontsize{16} x~{\chi}^2' '(4)'])                    %<8>
text(1.5,0.08,'\fontname{隶书}\fontsize{22}置信水平 0.9')             %<9>
hold off
```

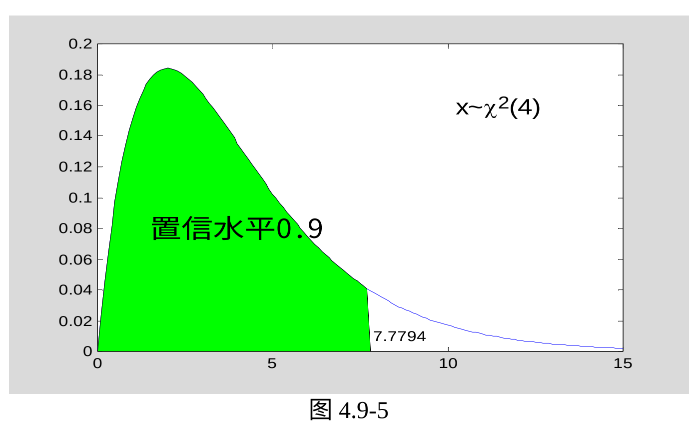


图 4.9-5

4.10多项式拟合和非线性最小二乘

4.10.1多项式拟合

4.10.1.1 多项式拟合原理

4.10.1.2 拟合多项式阶数的确定

4.10.1.3 多项式拟合的 MATLAB 实现

【例4.10.1.3-1】实施函数拟合的较完整描述示例。

```
%
x=0:0.1:1;y=[2.1,2.3,2.5,2.9,3.2,3.3,3.8,4.1,4.9,5.4,5.8];
dy=0.15;
for n=1:6
    [a,S]=polyfit(x,y,n);
    A{n}=a;
    da=dy*sqrt(diag(inv(S.R'*S.R)));
    DA{n}=da';
    freedom(n)=S.df;
    [ye,delta]=polyval(a,x,S);
    YE{n}=ye;
    D{n}=delta;
```

```
    chi2(n)=sum((y-ye).^2)/dy/dy;
end
Q=1-chi2cdf(chi2,freedom);
%
subplot(1,2,1),plot(1:6,abs(chi2-freedom),'b')
xlabel('阶次'),title('chi 2与自由度')
subplot(1,2,2),plot(1:6,Q,'r',1:6,ones(1,6)*0.5)
xlabel('阶次'),title('Q 与0.5 线')
```

图 4.10-1

```
%
clf,plot(x,y,'b+');axis([0,1,1,6]);hold on
errorbar(x,YE{3},D{3},'r');hold off
title('较适当的三阶拟合')
text(0.1,5.5,['chi2=' num2str(chi2(3)) '~' int2str(freedom(3))])
text(0.1,5,['freedom=' int2str(freedom(3))])
text(0.6,1.7,['Q=' num2str(Q(3)) '~0.5'])
```

图 4.10-2

```
A{3},DA{3}
ans =
    0.6993    1.2005    1.8869    2.1077
ans =
    1.9085    2.9081    1.2142    0.1333
```

4.10.2非线性最小二乘估计

4.10.2.1 伪线性最小二乘

4.10.2.2 借助 fminsearch 指令进行非线性最小二乘估计

【例 4.10.2.2-1】取发生信号的原始模型为 $y(x)=3e^{-0.4x}+12e^{-3.2x}$。$x$ 在[0, 4]中取值；y 受到噪声 0.3*(rand(n,1)-0.5) 的污染。本例演示：如何以 $y=a_1e^{-a_3x}+a_2e^{-a_4x}$ 为模型，通过 fminsearch 从受污染数据中，估计出参数 $a=[a(1),a(2),a(3),a(4)]=[a_1,a_2,a_3,a_4]$。

（1）

[xydata.m]

```
function [x,y,STDY]=xydata(k_noise)
%xydata.m
x=[0:0.2:4]';
yo=3*exp(-0.4*x)+12*exp(-3.2*x);
rand('seed',234)
y_noise=k_noise*(rand(size(x))-0.5);
y=yo+y_noise;
STDY=std(y_noise);
```

[twoexps.m]

```
function E=twoexps(a,x,y)
%twoexps.m
x=x(:);y=y(:);Y=a(1)*exp(-a(3)*x)+a(2)*exp(-a(4)*x);
E=sum((y-Y).^2);
```

（2）编写 M 脚本文件作为主文件

[exm041022_1.m]

```
%exm041022_1.m
k_noise=0.3;
[x,y,STDY]=xydata(k_noise);
a0=[1 1 1 1];
options=optimset('fminsearch');
options.TolX=0.01;
options.Display='off';
a=fminsearch(@twoexps,a0,options,x,y);
chi_est=twoexps(a,x,y)/STDY^2;
freedom=length(x)-length(a0);
Q=1-chi2cdf(chi_est,freedom);
%
y_est=a(1)*exp(-a(3)*x)+a(2)*exp(-a(4)*x);
ych='y_e_s_t=';
a1=num2str(a(1));a2=num2str(a(2));a3=num2str(a(3));a4=num2str(a(4));
char_y_est=[ych,a1,'*exp(-',a3,'*x) + ',a2,'*exp(-',a4,'*x)'];
plot(x,y,'b+');hold on,plot(x,y_est,'r');hold off,axis([0,4,0,16])
text(0.4,14,'y=3*exp(-0.4*x)+12*exp(-3.2*x)')
text(0.4,12,char_y_est),text(2.5,9,['chi2=' , num2str(chi_est)])
text(2.5,7,['freedom=' , num2str(freedom)])
text(2.5,5,['Q=' , num2str(Q)])
```

（3）

```
exm041022_1
```

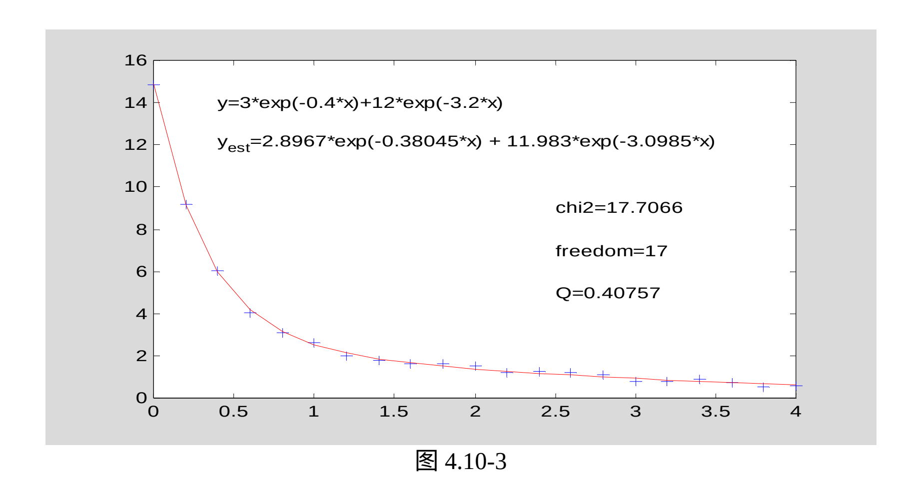


图 4.10-3

【例 4.10.2.2-2】以 $y=b_1e^{-a_1x}+b_2e^{-a_2x}$ 为模型，从受污染的数据中，使用伪线性法估计 $b=[b_1 \quad b_2]$，使用 fminsearch 估计 $a=[a_1 \quad a_2]$。x, y 原始数据同上例。

（1）

[twoexps2.m]

```
function E=twoexps2(a,x,y,b)
%twoexps2.m
x=x(:);y=y(:);Y=b(1)*exp(-a(1)*x)+b(2)*exp(-a(2)*x);
E=sum((y-Y).^2);
```

（2）

[exm041022_2.m]

```
%exm041022_2.m
k_noise=0.3;
[x,y,STDY]=xydata(k_noise);
a0=[1 2]';
options=optimset('fminsearch');
options.TolX=0.001;
options.Display='off';
%
while 1
   Mb=exp(-x*a0');
   b=Mb\y;
   a=fminsearch(@twoexps2,a0,options,x,y,b);
   r=norm(a-a0)/norm(a);
   if r<0.001;break;end
   a0=a;
end
chi_est=twoexps2(a,x,y,b)/STDY^2;
freedom=length(x)-length([a;b]);
Q=1-chi2cdf(chi_est,freedom);
%
y_est=b(1)*exp(-a(1)*x)+b(2)*exp(-a(2)*x);
ych='y_e_s_t=';
b1=num2str(b(1));b2=num2str(b(2));a1=num2str(a(1));a2=num2str(a(2));
char_y_est=[ych ,b1,'*exp(-',a1,'*x) + ',b2,'*exp(-',a2,'*x)'];
plot(x,y,'b+');hold on;plot(x,y_est,'r');hold off;axis([0,4,0,16])
```

```
text(0.4,14,'y=3*exp(-0.4*x)+12*exp(-3.2*x)');text(0.4,12,char_y_est)
text(2.5,9,['chi2=' , num2str(chi_est)])
text(2.5,7,['freedom=' , num2str(freedom)])
text(2.5,5,['Q=' , num2str(Q)])
```

（3）

```
exm041022_2
```

图 4.10-4

4.10.2.3 非线性最小二乘估计指令

【例 4.10.2.3-1】采用与例 4.10.2.2-1 相同的原始模型和受噪声污染的数据。运用 lsqnonlin 从受污染数据中，估计出 $y = a_1 e^{-a_3 x} + a_2 e^{-a_4 x}$ 的参数 $a = [a(1), a(2), a(3), a(4)] = [a_1, a_2, a_3, a_4]$。

（1）

[twoexps3.m]

```
function E=twoexps3(a,x,y)
x=x(:);y=y(:);Y=a(1)*exp(-a(3)*x)+a(2)*exp(-a(4)*x);
E=y-Y;
```

（2）

```
clear
k_noise=0.3;
[x,y,STDY]=xydata(k_noise);
a0=[1 10 0.2 1];
options=optimset('lsqnonlin');
options.TolX=0.01;options.Display='off';
a=lsqnonlin(@twoexps3,a0,[],[],options,x,y);
y_est=a(1)*exp(-a(3)*x)+a(2)*exp(-a(4)*x);
a1=num2str(a(1));a2=num2str(a(2));a3=num2str(a(3));a4=num2str(a(4));
char_y_est=['yest=',a1,'*exp(-',a3,'*x) + ',a2,'*exp(-',a4,'*x)'];
disp(['原方程  ', 'y=3*exp(-0.4*x)+12*exp(-3.2*x)'])
disp(['估计方程',char_y_est])
```

```
原方程  y=3*exp(-0.4*x)+12*exp(-3.2*x)
估计方程 yest=2.8381*exp(-0.37236*x) + 12.0338*exp(-3.0717*x)
```

4.11插值和样条

4.11.1 一维插值

【例 4.11.1-1】已知一组原始数据，确定它们所代表函数穿越 $y = 0.95$ 线的时刻。

（1）

```
t=linspace(0,5,100);y=1-cos(3*t).*exp(-t);
```

（2）

```
plot(t,y,'b');grid;hold on,plot(t,0.95*ones(size(t)),'r');hold off
```

图 4.11-1

```
t_ginput=ginput(1)
t_ginput =
    0.4965    0.9500
```

（3）

```
it=min(find(y>0.95));
T=(it-3):(it+3);                                                    %   <2>
t_nearst=interp1(y(T),t(T),0.95,'nearst');                          %   <3>
t_linear=interp1(y(T),t(T),0.95);                                %   <4>
t_cubic=interp1(y(T),t(T),0.95,'cubic');                            %   <5>
t_spline=interp1(y(T),t(T),0.95,'spline');                          %   <6>
disp(['  t_nearst  t_linear   t_cubic   t_spline'])
disp([t_nearst  t_linear   t_cubic   t_spline])
  t_nearst  t_linear   t_cubic   t_spline
    0.5051    0.4965    0.4962    0.4962
```

（4）

```
t_zero=fzero('1-cos(3*x).*exp(-x)-0.95',0.5)
t_zero =
    0.4962
```

4.11.2 高维函数的插值

【例 4.11.2-1】假设有一组海底深度测量数据，采用插值方式绘制海底形状图。

（1）

```
randn('state',2)
x=-5:5;y=-5:5;[X,Y]=meshgrid(x,y);
%
zz=1.2*exp(-((X-1).^2+(Y-2).^2))-0.7*exp(-((X+2).^2+(Y+1).^2));
Z=-500+zz+randn(size(X))*0.05;
surf(X,Y,Z);view(-25,25)
```

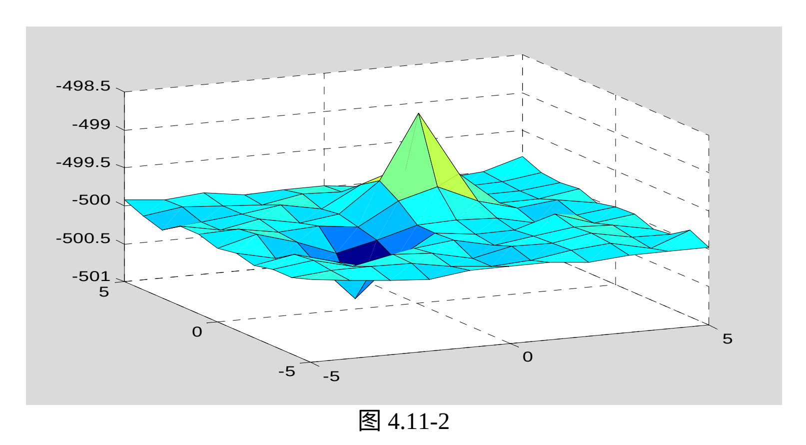


图 4.11-2

（2）

```
xi=linspace(-5,5,50);yi=linspace(-5,5,50);[XI,YI]=meshgrid(xi,yi);
ZI=interp2(X,Y,Z,XI,YI,'*cubic');
surf(XI,YI,ZI),view(-25,25)
```

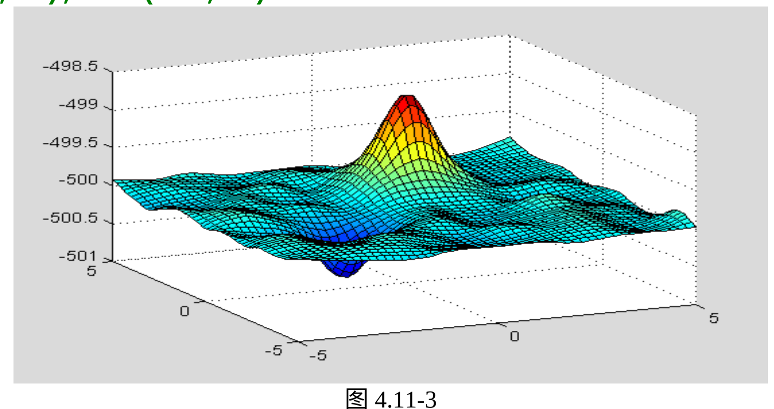


图 4.11-3

4.12样条函数及其应用

4.12.1样条插值

【例 4.12.1-1】根据连续时间函数 $w(t)=e^{-|t|}$ 的采样数据，利用 spline 重构该连续函数，并检查重构误差。

```
t=-5:0.5:5;w=exp(-abs(t));
N0=length(t);tt=linspace(t(1),t(end),10*N0);
ww=spline(t,w,tt);
error=max(abs(ww-exp(-abs(tt))))
plot(tt,ww,'b');hold on
stem(t,w,'filled','r');hold off
error =
    0.0840
```

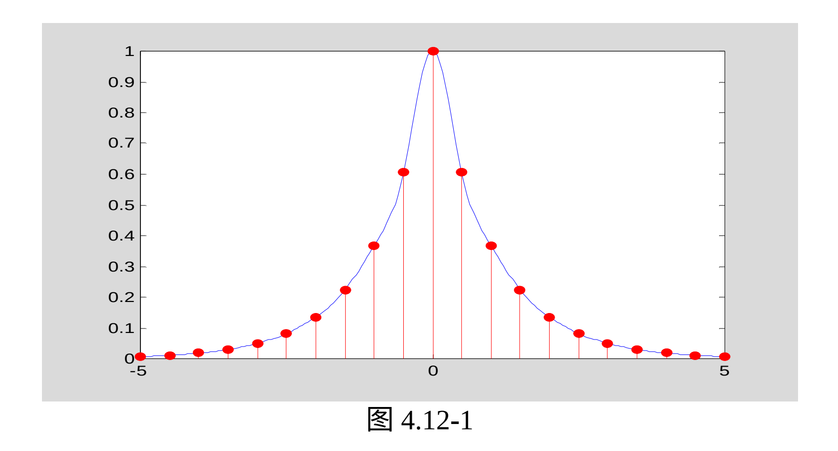


图 4.12-1

【例 4.12.1-2】用样条插值产生长、短轴分别在 45 度、135 度线上的椭圆。

```
theta=[0:0.5:2]*pi;
y=[-0.5 1 -0.5 -1 0.5 1 -0.5;0.5 1 0.5 -1 -0.5 1 0.5];  %<3>
theta2=linspace(theta(1),theta(end),50*length(theta));
yy=spline(theta,y,theta2);
plot(yy(1,:),yy(2,:),'b');hold on
plot(y(1,:),y(2,:),'or');hold off,axis('image')
```

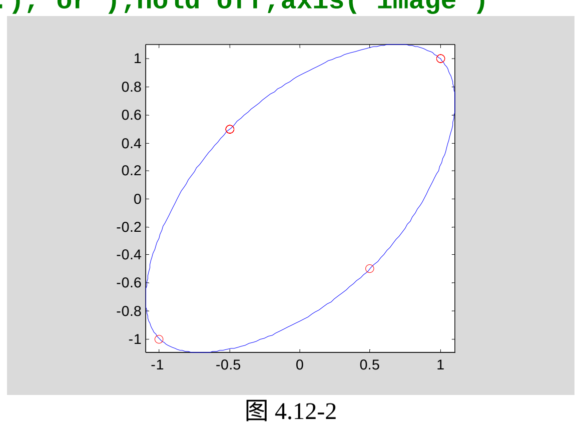


图 4.12-2

4.12.2样条函数用于数值积分和微分

【例 4.12.2-1】对于函数 $y = \sin x$，很容易求得 $S(x) = \int_0^x \sin xdx = 1 - \cos x$，$y' = \cos x$。本例将借此演示样条函数求数值不定积分、导函数的能力。

（1）

```
x=(0:0.1:1)*2*pi;y=sin(x);
pp=spline(x,y);
int_pp=fnint(pp);
der_pp=fnder(pp);
%
xx=(0:0.01:1)*2*pi;
err_yy=max(abs(ppval(pp,xx)-sin(xx)))
err_int=max(abs(ppval(int_pp,xx)-(1-cos(xx))))
err_der=max(abs(ppval(der_pp,xx)-cos(xx)))
err_yy =
    0.0026
err_int =
    0.0010
err_der =
```

```
    0.0253
```

(2)

```
%
DefiniteIntegral.bySpline=ppval(int_pp,[1,2])*[-1;1];        %   <2>
DefiniteIntegral.byTheory=(1-cos(2))-(1-cos(1));
%
Derivative.bySpline=fnval(der_pp,3);
Derivative.byTheory=cos(3);
Derivative.byDiference=(sin(3.01)-sin(3))/0.01;
DefiniteIntegral,Derivative
DefiniteIntegral =
    bySpline: 0.9563
    byTheory: 0.9564
Derivative =
       bySpline: -0.9895
       byTheory: -0.9900
    byDiference: -0.9907
```

(3)

```
fnplt(pp,'b-');hold on
fnplt(int_pp,'m:'),fnplt(der_pp,'r--');hold off
legend('y(x)','S(x)','dy/dx')
```

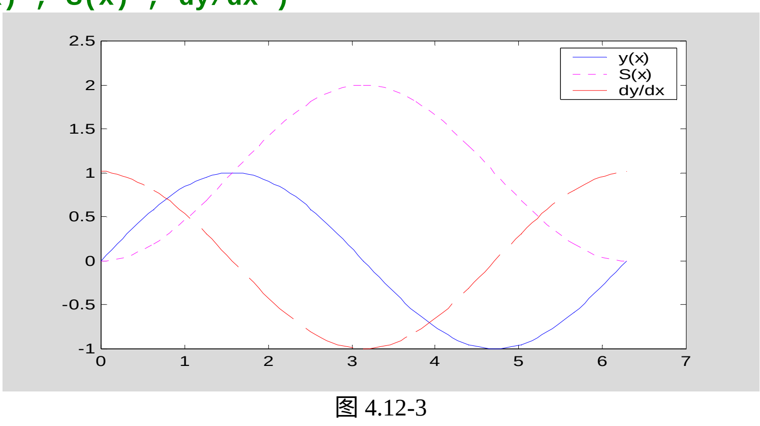


图 4.12-3

4.13Fourier 分析

4.13.1快速 Fourier 变换和逆变换指令

4.13.2连续时间函数的 Fourier 级数展开

4.13.2.1　展开系数的积分求取法

4.13.2.2　Fourier 级数与 DFT 之间的数学联系

4.13.2.3　MATLAB 算法实现

【例 4.13.2.3-1】已知时间函数 $w(t)=\begin{cases} t-0.5 & 0.5\le t\le 1.5 \\ 0 & else \end{cases}$，运用符号法求该函数的 Fourier 级数展开系数。

（1）

[fzzysym.m]

```
function [A_sym,B_sym]=fzzysym(T,Nf,Nn)
%
syms ttt n
if nargin<2;Nf=6;end
if nargin<3;Nn=32;end
yy=time_fun_s(ttt);
A0=int(yy,ttt,0,T)/T;
As=int(yy*cos(2*pi*n*ttt/T),ttt,0,T);
Bs=int(yy*sin(2*pi*n*ttt/T),ttt,0,T);
A_sym(1)=double(vpa(A0,Nn));
for k=1:Nf
   A_sym(k+1)=double(vpa(subs(As,n,k),Nn));
   B_sym(k+1)=double(vpa(subs(Bs,n,k),Nn));
end
%------------------------------------------
function yy=time_fun_s(ttt)
%
y1=sym('Heaviside(ttt-0.5)')*(ttt-0.5);
yy=y1-sym('Heaviside(ttt-1.5)')*((ttt-1.5)+1);
```

（2）

```
[A_sym,B_sym]=fzzysym(2)
A_sym =
    0.2500   -0.3183    0.0000    0.1061   -0.0000   -0.0637    0.0000
B_sym =
         0   -0.2026    0.1592    0.0225   -0.0796   -0.0081    0.0531
```

【例 4.13.2.3-2】运用数值积分法，按式(4.13.2.1-4)和(4.13.2.1-5)，求上例时间函数的 Fourier 级数展开系数。

（1）

[fzzyquad.m]

```
function [t,y,S,A,B]=fzzyquad(a,T,Nf,K)
%
%
if nargin<4;K=200;end
if (nargin<3|isempty(Nf));Nf=15;end
k=1:K;
t=a+k*T/length(k);
y=time_fun(t,T);
S=zeros(Nf+1,length(k));
a0=mean(y);
S(1,:)=a0;
A=zeros(1,Nf+1);B=zeros(1,Nf+1);
A(1)=a0;
for n = 1:Nf
   A(n+1) = quadl(@cos_y,a,a+T,[],[],n,T)/T*2;
   B(n+1) = quadl(@sin_y,a,a+T,[],[],n,T)/T*2;
   S(n+1,:)=S(n,:)+A(n+1)*cos(2*n*pi*t/T)+B(n+1)*sin(2*n*pi*t/T);
end
```

[cos_y.m]

```
function wcos=cos_y(t,n,T)
%
y=time_fun(t,T);wcos=cos(2*n*pi*t/T).*y;
```

[sin_y.m]

```
function wsin=sin_y(t,n,T)
%
y=time_fun(t,T);wsin=sin(2*n*pi*t/T).*y;
```

[time_fun.m]

```
function y=time_fun(t,T)
%
%
y=zeros(size(t));ii=find(t>=0.5 & t<=1.5);
y(ii)=ones(size(ii)).*(t(ii)-0.5);y(y==1.0)=0.5;
```

（2）

```
[t,y,S,aquad,bquad]=fzzyquad(0,2,15);   %  <1>
AA=abs(A_sym);AA(AA<1e-10)=NaN;
BB=abs(B_sym);BB(BB<1e-10)=NaN;
A_quad=aquad(1:7)
B_quad=bquad(1:7)
asq=abs(A_quad-A_sym)./AA
bsq=abs(B_quad-B_sym)./BB
A_quad =
    0.2500   -0.3183   -0.0000    0.1061    0.0000   -0.0637   -0.0000
B_quad =
         0   -0.2026    0.1592    0.0225   -0.0796   -0.0081    0.0531
asq =
  1.0e-005 *
         0    0.0105       NaN    0.0944       NaN    0.5316       NaN
bsq =
  1.0e-005 *
       NaN    0.4866    0.1281    0.2949    0.3402    0.8191    0.7655
```

（3）图形显示截断余项后三角展开近似波形

```
SS=[S;y]';ribbon(t,SS);
xlabel('谐波次数\rightarrow'),ylabel('t\rightarrow')
view([46,38]),colormap(jet),shading flat,light,lighting gouraud
```

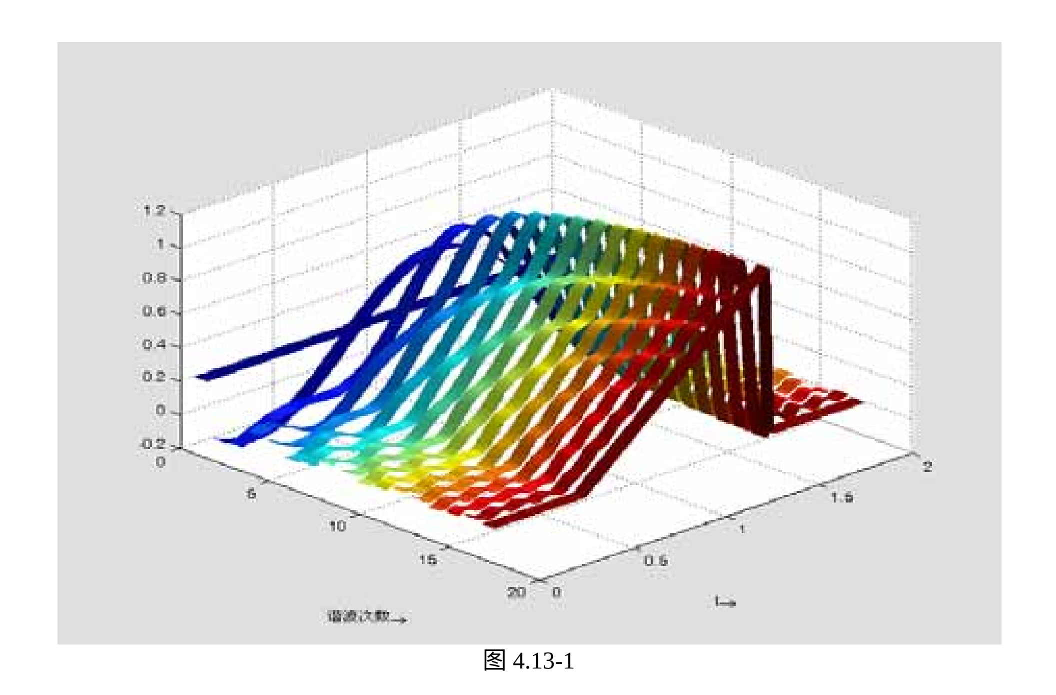


图 4.13-1

【例 4.13.2.3-3】运用 FFT，按式(4.13.2.2-2)和(4.13.2.2-3)，求上例时间函数的 Fourier 级数展开系数。

（1）

[fzzyfft.m]

```
function [A,B,C,fn,t,w]=fzzyfft(T,M,Nf)
%
if (nargin<2 | isempty(M));M=8;end                             %      <9>
if nargin<3;Nf=6;end                                           %      <10>
N=2^M;                      %
f=1/T;                      %
w0=2*pi*f;
dt=T/N;                     %
n=0:1:(N-1);                %
t=n*dt;                     %
w=time_fun(t,T);                                               %      <17>
%
W=fft(w);                   %
cn=W/N;                     %
%
z_cn=find(abs(cn)<1.0e-10);
cn(z_cn)=zeros(length(z_cn),1);                                %      <23>
cn_SH=fftshift(cn);                                            %
C=[cn_SH cn_SH(1)];         %
A(1)=C(N/2+1);
A(2:N/2+1)=2*real(C((N/2+2):end));
B(2:N/2+1)=-2*imag(C((N/2+2):end));
if Nf>N/2;error(['第三输入宗量 Nf 应小于 ' int2str(N/2-1)]);end
A(Nf+2:end)=[];
B(Nf+2:end)=[];
n1=-N/2:1:N/2;
fn=n1*f;
```

（2）运行以下指令

```
[A_fft,B_fft]=fzzyfft(2);
AA=abs(A_sym);AA(AA<1e-10)=NaN;BB=abs(B_sym);BB(BB<1e-10)=NaN;
ast=abs(A_sym-A_fft)./AA
bst=abs(B_sym-B_fft)./BB
ast =
         0    0.0001       NaN    0.0005       NaN    0.0013       NaN
bst =
       NaN    0.0001    0.0002    0.0005    0.0008    0.0013    0.0018
```

4.13.3利用 DFT 计算一般连续函数的 Fourier 变换 CFT

4.13.3.1　CFT 与 DFT 之间的数学联系

4.13.3.2　MATLAB 算法实现

【例 4.13.3.2-1】运用 FFT 求取矩形脉冲 $w(t)=\begin{cases}1 & 0\le t\le 1\\ 0 & else\end{cases}$ 的谱，说明采样频率低引起的混迭现象。

（1）

[cftbyfft.m]

```
function [AW,f]=cftbyfft(wt,t,flag)
%cftbyfft.m
if nargin==2;flag=1;end
N=length(t);                %
T=t(length(t))-t(1);        %
dt=T/N;                     %
W0=fft(wt);                 %       <16>
W=dt*W0;                    %
df=1/T;                     %
n=0:1:(N-1);
%
if flag==0
   n=-N/2:(N/2-1);
   W=fftshift(W);        %
end
f=n*df;                  %
AW=abs(W);               %
if nargout==0
    plot(f,AW);grid,xlabel('频率f');ylabel('|w(f)|')
end
```

（2）

```
M=5;       %      <1>
tend=1;    %
T=10;      %      <3>
N=2^M;     %
dt=T/N;    %
n=0:N-1;   %
t=n*dt;    %
w=zeros(size(t,2),1);
Tow=find((tend-t)>0);         %
```

```
w(Tow,1)=ones(length(Tow),1); %
plot(t,w,'b','LineWidth',2.5),title('Time Waveform');xlabel('t --- >')
```

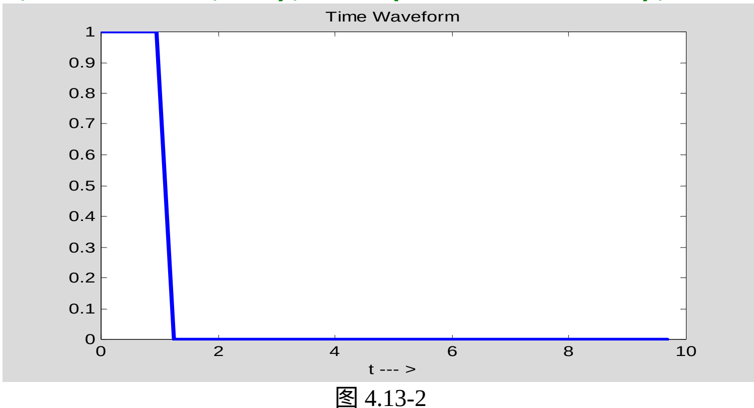


图 4.13-2

```
[AW,f]=cftbyfft(w,t,0);
ff=f+eps;
AWW=abs(sin(pi*ff)./(pi*ff));
plot(f,AW,'b-',ff,AWW,'r:')
title('Aliasing caused by undersampling')
xlabel('f --- >');ylabel('|W(f)|'),legend('by FFT','Theoretical')
```

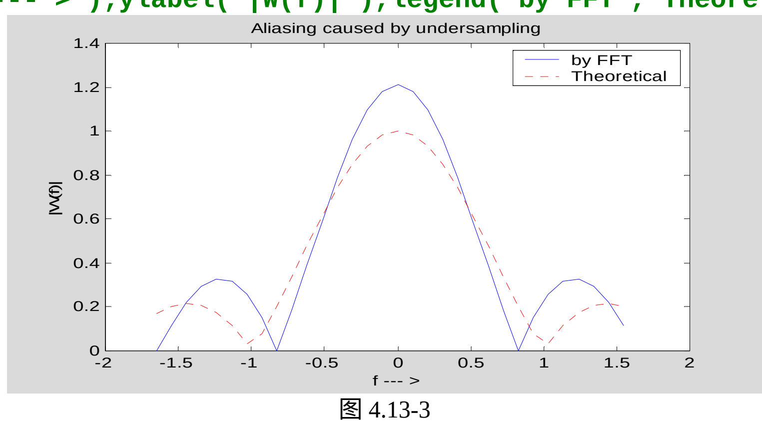


图 4.13-3

4.14 常微分方程

4.14.1 初值常微分方程的解算指令

4.14.1.1 解 ODE 的基本机理

4.14.1.2 solver 解算指令的使用说明

4.14.2 ODE 解算指令的使用演示

4.14.2.1 解算指令简洁格式使用示例

【例 4.14.2.1-1】采用最简洁格式的 ODE 文件和解算指令，研究围绕地球旋转的卫星轨道。

（1）

（2）

（3）

[dYdt.m]

```
function Yd=DYdt(t,Y)
% t
% Y
global G ME                  %
xy=Y(1:2);Vxy=Y(3:4);        %
r=sqrt(sum(xy.^2));
Yd=[Vxy;-G*ME*xy/r^3];       %
```

（4）

```
global G ME                   %      <1>
G=6.672e-11;ME=5.97e24;vy0=4000;x0=-4.2e7;t0=0;tf=60*60*24*9;
tspan=[t0,tf];                %
Y0=[x0;0;0;vy0];              %
[t,YY]=ode45('DYDt',tspan,Y0);%      <8>
X=YY(:,1);                    %
Y=YY(:,2);                    %
plot(X,Y,'b','Linewidth',2); hold on
axis('image')                 %
[XE,YE,ZE] = sphere(10);      %
RE=0.64e7;                    %
XE=RE*XE;YE=RE*YE;ZE=0*ZE;    %
mesh(XE,YE,ZE),hold off       %
```

图 4.14-2

【例 4.14.2.1-2】上例中，程序间的参数（如 G 和 ME）传送，是依靠全局变量形式实现的。一般说来，编写程序时，应尽量少用全局变量，以免引起混乱。本例演示参数如何在指令间直接传送。

（1）

[DYDt2.m]

```
function Yd=DYDt2(t,Y,flag,G,ME)
% flag
%
switch flag
case ''          %
   X=Y(1:2);V=Y(3:4);r=sqrt(sum(X.^2));Yd=[V;-G*ME*X/r^3];
```

```
otherwise
  error(['Unknown flag ''' flag '''.']);
end
```

(2)

```
[t,YY]=ode45('DYDt2',tspan,Y0,[],G,ME);
```

4.14.2.2 解算指令较复杂格式的使用示例

【例 4.14.2.2-1】带事件设置的 ODE 文件及主程序编写演示。本例将以较高精度计算卫星经过近地点和远地点的时间，并在图上标志。

(1)

[DYDt3.m]

```
function varargout=DYDt3(t,Y,flag,G,ME,tspan,Y0)
%
%
%
%
%
%
%
switch flag
case ''        %
  varargout{1} = f(t,Y,G,ME);      %
case 'init'   %
  [varargout{1:3}] = fi(tspan,Y0);%
case 'events' %
  [varargout{1:3}] = fev(t,Y,Y0);
otherwise
  error(['Unknown flag ''' flag '''.']);
end
% ------------------------------------------------------------------
function Yd = f(t,Y,G,ME)
%
X=Y(1:2);V=Y(3:4);r=sqrt(sum(X.^2));Yd=[V; -G*ME*X/r^3];
% ------------------------------------------------------------------
function [ts,y0,options] = fi(tspan,Y0)
%
ts=tspan;y0 = Y0;
options = odeset('Events','on','Reltol',1e-5,'Abstol',1e-4);
                 %
% ------------------------------------------------------------------
function [value,isterminal,direction] = fev(t,Y,Y0)
%
dDSQdt = 2 * ((Y(1:2)-Y0(1:2))' * Y(3:4));
%dDSQdt
value = [dDSQdt; dDSQdt];   %
direction = [1; -1]; %
isterminal = [1; 0]; %
```

(2)

```
G=6.672e-11;ME=5.97e24;vy0=4000; x0=-4.2e7;t0=0;tf=60*60*24*9;
tspan=[t0,tf];Y0=[x0;0;0;vy0];
[t,YY,Te,Ye,Ie]=ode45('DYDt3',[],[],[],G,ME,tspan,Y0);   %     <3>
X=YY(:,1);Y=YY(:,2);
plot(X,Y,'b','Linewidth',2);hold on
text(0,6e7,'轨道','Color','b')
axis('image');
%
plot(Ye(1,1),0.4e7+Ye(1,2),'r^','MarkerSize',10)
plot(Ye(2,1),0.4e7+Ye(2,2),'bv','MarkerSize',10)
plot(Ye(3,1),-0.4e7+Ye(3,2),'b^','MarkerSize',10)
%
text(0.8*Ye(3,1),-2e7+Ye(3,2),['t3=' sprintf('%6.0f',Te(3))])
text(0.8*Ye(2,1),1.5e7+Ye(2,2),['t2=' sprintf('%6.0f',Te(2))])
%
[XE,YE,ZE] = sphere(10);RE=0.64e7;XE=RE*XE;YE=RE*YE;ZE=0*ZE;
mesh(XE,YE,ZE)
text(1e7,1e7,'地球','Color','r'), hold off
```

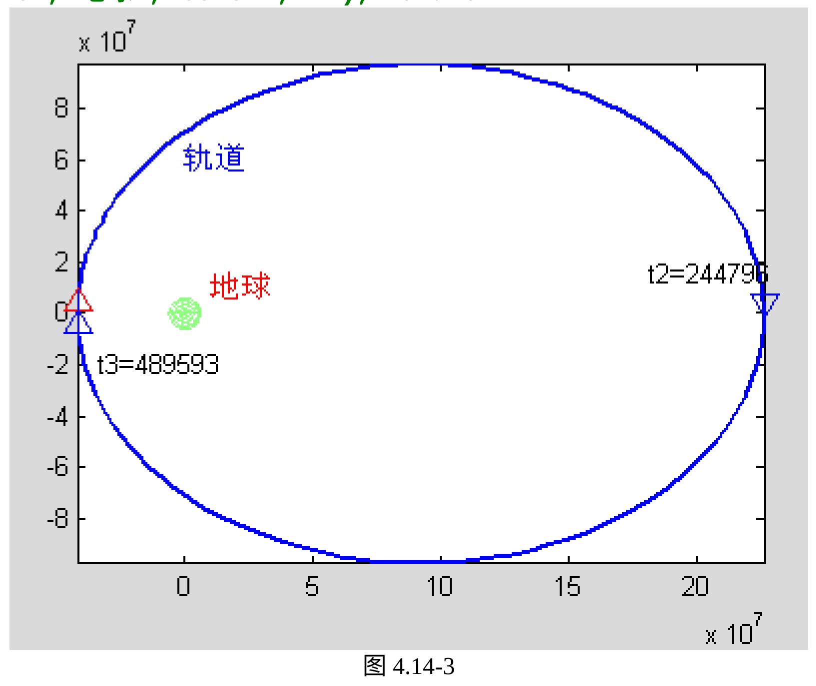


图 4.14-3

4.14.3 关于 ODE 文件的说明

4.14.4 关于解算指令选项 options 的属性设置

4.14.4.1　options 的属性域名

4.14.4.2　options 属性处理和输出函数使用演示

【例 4.14.4.2-1】仍以卫星轨道问题为例（原题见例 4.14.2.1-1, 4.14.2.1-2，4.14.2.2-1）。本例演示如何通过对 options 域的直接设置，借助微分方程解算输出指令，表现解算的中间结果。具体目标是：画出解向量 $Y=[y_1 \quad y_2 \quad y_3 \quad y_4]^T=[x \quad y \quad v_x \quad v_y]^T$ 中由 x, v_x 构成的相平面。相平面的绘制是在微分方程解算中间逐步完成的。

（1）

[DYDt4.m]

```
function varargout=DYDt4(t,Y,flag,G,ME,tspan,Y0)
switch flag
case ''
   varargout{1} = f(t,Y,G,ME);
case 'init'
  [varargout{1:3}] = fi(tspan,Y0);
otherwise
  error(['Unknown flag ''' flag '''.']);
end
% ------------------------------------------------------------------
function Yd = f(t,Y,G,ME)
X=Y(1:2);V=Y(3:4);r=sqrt(sum(X.^2));Yd=[V; -G*ME*X/r^3];
% ------------------------------------------------------------------
function [ts,y0,options] = fi(tspan,Y0)
ts=tspan;y0 = Y0;
%
options.RelTol=1e-5;options.AbsTol=1e-4;
options.OutputFcn='odephas2';
options.OutputSel=[1 3];
```

（2）

[odeexp4.m]

```
%odeexp4.m
G=6.672e-11;ME=5.97e24;vy0=4000;x0=-4.2e7;t0=0;tf=60*60*24*9;
tspan=[t0,tf];Y0=[x0;0;0;vy0];
%
clf,set(gca,'xlim',[-5 25]*1e7,'ylim',[-3 3]*1e3);%    <5>
box on
hold on;
ode45('DYDt4',[],[],[],G,ME,tspan,Y0);hold off
```

(3)

```
shg,odeexp4
```

图 4.14-4

4.14.5 常微分方程的边值问题解

4.14.5.1 bvp4c 求解边值问题的基本思路

4.14.5.2 求解边值问题的基本配套指令

【例 4.14.5.2-1】求二阶方程 $z''+c\cdot|z|=0$ 满足边界条件 $z(0)=0, z(4)=-2$ 的解。在此，取 c=1。本例的目的：（A）完整演示解题步骤；（B）微分方程和边界条件的标准写法；（C）导数函数文件和边界残差函数文件的编写；（D）已知参数的传递；（E）初始猜测网、近似解、插值解的形态。

（1）

（2）

[twoode.m]

```
function  dydx=twoode(x,y,c)
dydx=[ y(2) , -c*abs(y(1))]';
```

[twobc.m]

```
function res = twobc(ya,yb,c)
%
%
res=[ ya(1), yb(1)+2]';
```

（3）

```
sinit=bvpinit(linspace(0,4,5),[1;0]);
```

（4）

```
c=1;
sol=bvp4c(@twoode,@twobc,sinit,[ ],c);                    %<3>
```

（5）

```
x=linspace(0,4,100);
y = bvpval(sol,x);
plot(x,y(1,:),'b-',sol.x,sol.y(1,:),'ro',sinit.x,sinit.y(1,:),'ks')
legend('\fontname{隶书}\fontsize{20}插值后的解曲线','解点','猜测解点',0)
```

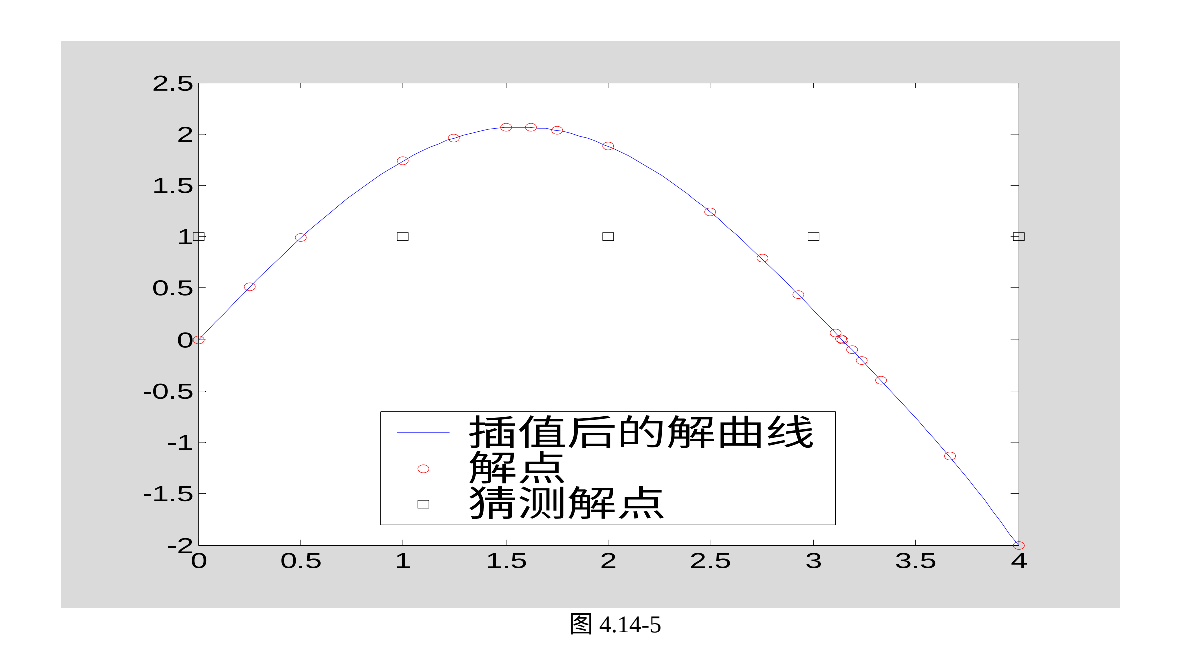


图 4.14-5

4.14.5.3　改善边值问题的求解性能

【例 4.14.5.3-1】求解微分方程 $z''+(\lambda-2q\cos 2x)z=0$，在此令 $q=15$。边界条件为 $z(0)=1, z'(0)=0, z'(\pi)=0$。本例的目的：（A）演示含未知参数微分方程边值问题的求解；（B）边界条件的数学表达和 M 函数文件的编写；（C）函数型猜测解的构成方式；（D）使用 bvpset 改变选项属性；（E）采用构架直接赋值法改变选项属性。（F）演示“接续”求解法。

（1）

（2）

[mat4ode.m]

```
function  dydx=mat4ode(x,y,lmb)
q=5;
dydx=[y(2); -(lmb – 2*q*cos(2*x))*y(1)];
```

[mat4bc.m]

```
function  res=mat4bc(ya,yb,lmb)
res=[ ya(1)-1 ; ya(2) ; yb(2) ];
```

（3）

```
clear
x00=linspace(0,pi,4);                           %
y00=inline('[cos(4*x);-4*sin(4*x)]');           %  <3>
lmb=15;                                         %
s0=bvpinit(x00,y00,lmb);                        %
```

（4）

```
opts=bvpset('AbsTol',0.5,'RelTol',0.38,'Stats','on');%  <6>
s1= bvp4c(@mat4ode,@mat4bc,s0,opts);            %
Lambda1=s1.parameters
The solution was obtained on a mesh of 7 points.
The maximum residual is 1.263e-001.
There were 356 calls to the ODE function.
There were 67 calls to the BC function.
Lambda1 =
```

```
   18.0803
```

(5)

```
opts.AbsTol=1e-6;opts.RelTol=1e-3;
s2= bvp4c(@mat4ode,@mat4bc,s1,opts);
Lambda2=s2.parameters
The solution was obtained on a mesh of 39 points.
The maximum residual is 4.915e-004.
There were 1443 calls to the ODE function.
There were 52 calls to the BC function.
Lambda2 =
   17.0973
```

(6)

```
plot(s0.x,s0.y(1,:),'ks--',s1.x,s1.y(1,:),'bo:',s2.x,s2.y(1,:),'r*-')
legend('\fontname{隶书}\fontsize{16}猜测解','第一近似解','第二近似解',0)
axis([0,pi,-1,1.2]),xlabel('x'),ylabel('solution y')
```

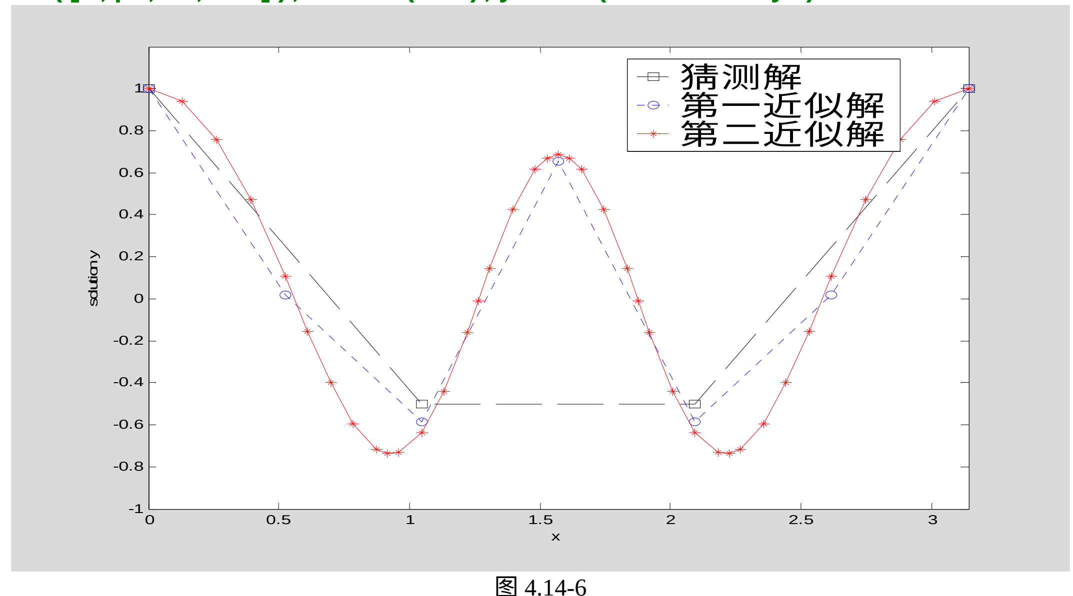


图 4.14-6

4.15 稀疏矩阵

4.15.1稀疏矩阵的存储方式

4.15.2稀疏矩阵的创建

4.15.2.1　稀疏矩阵创建指令：sparse

4.15.2.2　稀疏带状矩阵创建指令：spdiags

4.15.2.3　外部数据转换为稀疏矩阵的指令：spconvert

【例 4.15.2.3-1】用两种不同方式创建三对角稀疏矩阵。

```
n=5;SM1=sparse(1:n,1:n,-2*ones(1,n),n,n,n);
```

```
SM2=sparse(2:n,1:n-1,ones(1,n-1),n,n,n-1);S1=SM1+SM2+SM2'
e=ones(n,1);S2=spdiags([e,-2*e,e],[-1,0,1],n,n),SF=full(S1)
S1 =
   (1,1)       -2
   (2,1)        1
   (1,2)        1
   (2,2)       -2
   (3,2)        1
   (2,3)        1
   (3,3)       -2
   (4,3)        1
   (3,4)        1
   (4,4)       -2
   (5,4)        1
   (4,5)        1
   (5,5)       -2
S2 =
   (1,1)       -2
   (2,1)        1
   (1,2)        1
   (2,2)       -2
   (3,2)        1
   (2,3)        1
   (3,3)       -2
   (4,3)        1
   (3,4)        1
   (4,4)       -2
   (5,4)        1
   (4,5)        1
   (5,5)       -2
SF =
    -2     1     0     0     0
     1    -2     1     0     0
     0     1    -2     1     0
     0     0     1    -2     1
     0     0     0     1    -2
```

4.15.3稀疏矩阵的运算

4.15.3.1 基本规则

4.15.3.2 常用指令及应用举例

【例 4.15.3.2-1】全元素矩阵、稀疏矩阵、最小排序稀疏矩阵三角分解所需时间的比较。

```
clear all,n=200;                 %
rand('state',1),randn('state',2) %
A=sprandsym(n,0.015,0.1,1);      %
subplot(1,2,1),spy(A,'b',10),title('Spy plot of matrix A')
subplot(1,2,2),d=symmmd(A);      %
spy(A(d,d),'b',10),title('Matrix A with Minimun degree ordering');
B=full(A);                       %
%
format short e
tic, L1=chol(B);t1=toc;          %
tic, L2=chol(A);t2=toc/t1;       %
tic, L3=chol(A(d,d));t3=toc/t1;  %
disp('    全元素阵      稀疏矩阵         最小排序阵'),disp([1,t2,t3])
    全元素阵      稀疏矩阵        最小排序阵
   1             0              0
```

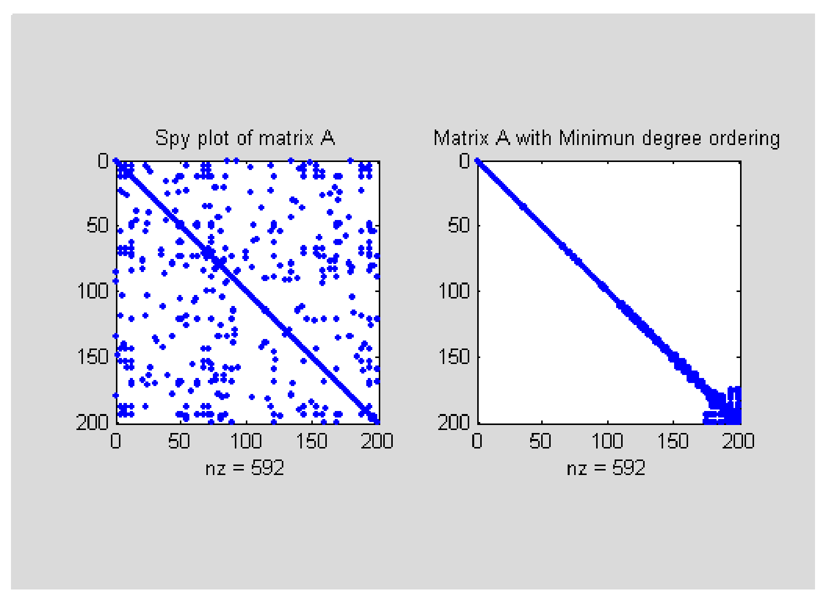

Spy plot of matrix A
0
50
100
150
200
0
50
100
150
200
nz = 592
Matrix A with Minimun degree ordering
0
50
100
150
200
0
50
100
150
200
nz = 592

图 4.15-1　稀疏结构

第五章　符号计算

符号计算的特点：一，运算以推理解析的方式进行，因此不受计算误差积累问题困扰；二，符号计算，或给出完全正确的封闭解，或给出任意精度的数值解（当封闭解不存在时）；三，符号计算指令的调用比较简单，经典教科书公式相近；四，计算所需时间较长，有时难以忍受。

在 MATLAB 中，符号计算虽以数值计算的补充身份出现，但涉及符号计算的指令使用、运算符操作、计算结果可视化、程序编制以及在线帮助系统都是十分完整、便捷的。

MATLAB 的升级和符号计算内核 Maple 的升级，决定着符号计算工具包的升级。但从用户使用角度看，这些升级所引起的变化相当细微。即使这样，本章还是及时作了相应的更新和说明。如 MATLAB 6.5+ 版开始启用 Maple VIII 的计算引擎，从而克服了 Maple V 计算“广义 Fourier 变换”时的错误（详见第 5.4.1 节）。

5.1 符号对象和符号表达式

5.1.1 符号对象的生成和使用

【例 5.1.1-1】符号常数形成中的差异

```
a1=[1/3,pi/7,sqrt(5),pi+sqrt(5)]           % <1>
a2=sym([1/3,pi/7,sqrt(5),pi+sqrt(5)])      % <2>
a3=sym([1/3,pi/7,sqrt(5),pi+sqrt(5)],'e')  % <3>
a4=sym('[1/3,pi/7,sqrt(5),pi+sqrt(5)]')    % <4>
a24=a2-a4
a1 =
    0.3333    0.4488    2.2361    5.3777
a2 =
[                   1/3,                     pi/7,                    sqrt(5),
6054707603575008*2^(-50)]
a3 =
[           1/3-eps/12,        pi/7-13*eps/165,     sqrt(5)+137*eps/280,
6054707603575008*2^(-50)]
a4 =
[         1/3,        pi/7,     sqrt(5), pi+sqrt(5)]
a24 =
[                     0,                                          0,
                      0, 189209612611719/35184372088832-pi-5^(1/2)]
```

【例 5.1.1-2】演示：几种输入下产生矩阵的异同。

```
a1=sym([1/3,0.2+sqrt(2),pi])        % <1>
a2=sym('[1/3,0.2+sqrt(2),pi]')      % <2>
a3=sym('[1/3  0.2+sqrt(2)  pi]')    % <3>
a1_a2=a1-a2                         %
a1 =
[                  1/3, 7269771597999872*2^(-52),                  pi]
a2 =
[          1/3, 0.2+sqrt(2),          pi]
a3 =
[          1/3, 0.2+sqrt(2),          pi]
a1_a2 =
[ 0, 1.4142135623730951010657008737326-2^(1/2),                     0]
```

【例 5.1.1-3】把字符表达式转换为符号变量

```
y=sym('2*sin(x)*cos(x)')
y=simple(y)
y =
2*sin(x)*cos(x)
y =
sin(2*x)
```

【例 5.1.1-4】用符号计算验证三角等式 $\sin\varphi_1\cos\varphi_2-\cos\varphi_1\sin\varphi_2=\sin(\varphi_1-\varphi_2)$。

```
syms fai1 fai2;y=simple(sin(fai1)*cos(fai2)-cos(fai1)*sin(fai2))
y =
sin(fai1-fai2)
```

【例 5.1.1-5】求矩阵 $A=\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ 的行列式值、逆和特征根

```
syms a11 a12 a21 a22;A=[a11,a12;a21,a22]
DA=det(A),IA=inv(A),EA=eig(A)
A =
[ a11, a12]
[ a21, a22]
DA =
a11*a22-a12*a21
IA =
[  a22/(a11*a22-a12*a21), -a12/(a11*a22-a12*a21)]
[ -a21/(a11*a22-a12*a21),  a11/(a11*a22-a12*a21)]
EA =
[ 1/2*a11+1/2*a22+1/2*(a11^2-2*a11*a22+a22^2+4*a12*a21)^(1/2)]
[ 1/2*a11+1/2*a22-1/2*(a11^2-2*a11*a22+a22^2+4*a12*a21)^(1/2)]
```

【例 5.1.1-6】验证积分 $\int_{-\tau/2}^{\tau/2} Ae^{-i\omega t}dt = A\tau\cdot\frac{\sin\frac{\omega\tau}{2}}{\frac{\omega\tau}{2}}$。

```
syms A t tao w;yf=int(A*exp(-i*w*t),t,-tao/2,tao/2);Yf=simple(yf)
Yf =
2*A*sin(1/2*tao*w)/w
```

5.1.2 符号计算中的算符和基本函数

5.1.3 识别对象类别的指令

【例 5.1.3-1】数据对象及其识别指令的使用。

（1）

```
clear,a=1;b=2;c=3;d=4;
Mn=[a,b;c,d]
Mc='[a,b;c,d]'
Ms=sym(Mc)
Mn =
     1     2
     3     4
Mc =
[a,b;c,d]
Ms =
```

```
[ a, b]
[ c, d]
```

(2)

```
SizeMn=size(Mn),SizeMc=size(Mc),SizeMs=size(Ms)
SizeMn =
     2     2
SizeMc =
     1     9
SizeMs =
     2     2
```

(3)

```
CMn=class(Mn),CMc=class(Mc),CMs=class(Ms)
CMn =
double
CMc =
char
CMs =
sym
```

(4)

```
isa(Mn,'double'),isa(Mc,'char'),isa(Ms,'sym')
ans =
     1
ans =
     1
ans =
     1
```

(5)

```
whos Mn Mc Ms
  Name      Size           Bytes  Class

  Mc        1x9               18  char array
  Mn        2x2               32  double array
  Ms        2x2              408  sym object

Grand total is 21 elements using 458 bytes
```

5.1.4 符号表达式中自由变量的确定

【例 5.1.4-1】对独立自由符号变量的自动辨认。

(1)

```
syms a b x X Y;k=sym('3');z=sym('c*sqrt(delta)+y*sin(theta)');
EXPR=a*z*X+(b*x^2+k)*Y;
```

(2)

```
findsym(EXPR)
ans =
X, Y, a, b, c, delta, theta, x, y
```

(3)

```
findsym(EXPR,1)
ans =
x
```

(4)

```
findsym(EXPR,2),findsym(EXPR,3)
ans =
x,y
```

```
ans =
x,y,theta
```

【例 5.1.4-2】findsym 确定自由变量是对整个矩阵进行的。

```
syms a b t u v x y;A=[a+b*x,sin(t)+u;x*exp(-t),log(y)+v]
findsym(A,1)
A =
[      a+b*x,   sin(t)+u]
[ x*exp(-t),   log(y)+v]
ans =
x
```

5.2 符号表达式和符号函数的操作

5.2.1 符号表达式的操作

【例 5.2.1-1】按不同的方式合并同幂项。

```
EXPR=sym('(x^2+x*exp(-t)+1)*(x+exp(-t))');
expr1=collect(EXPR)
expr2=collect(EXPR,'exp(-t)')
expr1 =
x^3+2*exp(-t)*x^2+(1+exp(-t)^2)*x+exp(-t)
expr2 =
x*exp(-t)^2+(2*x^2+1)*exp(-t)+(x^2+1)*x
```

【例 5.2.1-2】factor 指令的使用

（1）

```
syms a x;f1=x^4-5*x^3+5*x^2+5*x-6;factor(f1)
ans =
(x-1)*(x-2)*(x-3)*(x+1)
```

（2）

```
f2=x^2-a^2;factor(f2)
ans =
(x-a)*(x+a)
```

（3）

```
factor(1025)
ans =
     5     5    41
```

【例 5.2.1-3】对多项式进行嵌套型分解

```
clear;syms a x;f1=x^4-5*x^3+5*x^2+5*x-6;horner(f1)
ans =
-6+(5+(5+(-5+x)*x)*x)*x
```

【例 5.2.1-4】写出矩阵$\begin{bmatrix} \frac{3}{2} & \frac{x^2+3}{2x-1}+\frac{3x}{x-1} \\ \frac{4}{x^2} & 3x+4 \end{bmatrix}$各元素的分子、分母多项式

(1)

```
syms x;A=[3/2,(x^2+3)/(2*x-1)+3*x/(x-1);4/x^2,3*x+4];
[n,d]=numden(A)
pretty(simplify(A))   %<3>
n =
[            3, x^3+5*x^2-3]
[            4,       3*x+4]
d =
[              2, (2*x-1)*(x-1)]
[            x^2,             1]

                            [              3       2        ]
                            [             x  + 5 x  - 3     ]
                            [3/2      -----------------]
                            [         (2 x - 1) (x - 1)]
                            [                               ]
                            [ 4                             ]
                            [----          3 x + 4          ]
                            [  2                            ]
                            [ x                             ]
```

(2)

```
pretty(simplify(n./d))
```

【例 5.2.1-5】简化 $f=\sqrt[3]{\frac{1}{x^3}+\frac{6}{x^2}+\frac{12}{x}+8}$

(1)

```
syms x;f=(1/x^3+6/x^2+12/x+8)^(1/3);
sfy1=simplify(f),sfy2=simplify(sfy1)
sfy1 =
((2*x+1)^3/x^3)^(1/3)
sfy2 =
((2*x+1)^3/x^3)^(1/3)
```

(2)

```
g1=simple(f),g2=simple(g1)
g1 =
(2*x+1)/x
g2 =
2+1/x
```

【例 5.2.1-6】简化 $ff=\cos x+\sqrt{-\sin^2 x}$

```
syms x;ff=cos(x)+sqrt(-sin(x)^2);
ssfy1=simplify(ff),ssfy2=simplify(ssfy1)
ssfy1 =
cos(x)+(-sin(x)^2)^(1/2)
ssfy2 =
cos(x)+(-sin(x)^2)^(1/2)

gg1=simple(ff),gg2=simple(gg1)
gg1 =
cos(x)+i*sin(x)
gg2 =
```

```
exp(i*x)
```

5.2.2 符号函数的求反和复合

【例 5.2.2-1】求 $f=x^2$ 的反函数

```
syms x;f=x^2;g=finverse(f)
Warning: finverse(x^2) is not unique.
> In D:\MATLAB6P5\toolbox\symbolic\@sym\finverse.m at line 43
g =
x^(1/2)
fg=simple(compose(g,f))       %验算 g(f(x))是否等于 x
fg =
x
```

【例 5.2.2-2】求 $f=\dfrac{x}{1+u^2}, g=\cos(y+fai)$ 的复合函数

（1）

```
syms x y u fai t;f=x/(1+u^2);g=cos(y+fai);fg1=compose(f,g)
fg1 =
cos(y+fai)/(1+u^2)
```

（2）

```
fg2=compose(f,g,u,fai,t)
fg2 =
x/(cos(y+t)^2+1)
```

5.2.3 置换及其应用

5.2.3.1 自动执行的子表达式置换指令

【例 5.2.3.1-1】演示子表达式的置换表示。

```
clear all,syms a b c d W;[V,D]=eig([a b;c d]);
[RVD,W]=subexpr([V;D],W)                                       %<2>
RVD =
[ -(1/2*d-1/2*a-1/2*W)/c, -(1/2*d-1/2*a+1/2*W)/c]
[                       1,                      1]
[      1/2*d+1/2*a+1/2*W,                       0]
[                       0,      1/2*d+1/2*a-1/2*W]
W =
(d^2-2*a*d+a^2+4*b*c)^(1/2)
```

5.2.3.2 通用置换指令

【例 5.2.3.2-1】用简单算例演示 subs 的置换规则。

（1）

```
syms a x;f=a*sin(x)+5;
```

（2）

```
f1=subs(f,'sin(x)',sym('y'))                                   %<2>
f1 =
a*y+5
```

（3）
```
f2=subs(f,{a,x},{2,sym(pi/3)})                                %<3>
f2 =
3^(1/2)+5
```

（4）
```
f3=subs(f,{a,x},{2,pi/3})                                     %<4>
f3 =
    6.7321
```

（5）
```
f4=subs(subs(f,a,2),x,0:pi/6:pi)                              %<5>
f4 =
    5.0000    6.0000    6.7321    7.0000    6.7321    6.0000    5.0000
```

（6）
```
f5=subs(f,{a,x},{0:6,0:pi/6:pi})                              %<6>
f5 =
    5.0000    5.5000    6.7321    8.0000    8.4641    7.5000    5.0000
```

5.2.4 符号数值精度控制和任意精度计算

5.2.4.1 向双精度数值转换的 doblue 指令

5.2.4.2 任意精度的符号数值

【例 5.2.4.2-1】指令使用演示。
```
digits
 Digits = 32

p0=sym('(1+sqrt(5))/2');
p1=sym((1+sqrt(5))/2)
e01=vpa(abs(p0-p1))
p1 =
7286977268806824*2^(-52)
e01 =
.543211520368250e-16

p2=vpa(p0)
e02=vpa(abs(p0-p2),64)
p2 =
1.6180339887498948482045868343657
e02 =
.6188227969082019423713786455137 7e-31

digits
 Digits = 32
```

5.2.5 符号对象与其它数据对象间的转换

【例 5.2.5-1】符号、数值间的转换。
```
phi=sym((1+sqrt(5))/2)
double(phi)
phi =
7286977268806824*2^(-52)
```

```
ans =
    1.6180
```

【例 5.2.5-2】各种多项式表示形式之间的转换

```
syms x;f=x^3+2*x^2-3*x+5;
sy2p=sym2poly(f)
p2st=poly2str(sy2p,'x')
p2sy=poly2sym(sy2p)
pretty(f,'x')
sy2p =
     1     2    -3     5
p2st =
   x^3 + 2 x^2 - 3 x + 5
p2sy =
x^3+2*x^2-3*x+5

                               3      2
                              x  + 2 x  - 3 x + 5
```

5.3 符号微积分

5.3.1 符号序列的求和

【例 5.3.1-1】求 $\sum_{t=0}^{t-1}\begin{bmatrix} t & k^3 \end{bmatrix}$，$\sum_{k=1}^{\infty}\begin{bmatrix} \frac{1}{(2k-1)^2} & \frac{(-1)^k}{k} \end{bmatrix}$

```
syms k t;f1=[t k^3];f2=[1/(2*k-1)^2,(-1)^k/k];
s1=simple(symsum(f1))
s2=simple(symsum(f2,1,inf))
s1 =
[ 1/2*t*(t-1),        k^3*t]
s2 =
[ 1/8*pi^2,   -log(2)]
```

5.3.2 符号微分和 *jacobian* 矩阵

【例 5.3.2-1】求 $\frac{d}{dx}\begin{bmatrix} a & t^3 \\ t\cos x & \ln x \end{bmatrix}$、$\frac{d^2}{dt^2}\begin{bmatrix} a & t^3 \\ t\cos x & \ln x \end{bmatrix}$和$\frac{d^2}{dxdt}\begin{bmatrix} a & t^3 \\ t\cos x & \ln x \end{bmatrix}$

```
syms a t x;f=[a,t^3;t*cos(x), log(x)];
df=diff(f)
dfdt2=diff(f,t,2)
dfdxdt=diff(diff(f,x),t)
df =
[          0,          0]
[ -t*sin(x),         1/x]
dfdt2 =
[    0, 6*t]
[    0,    0]
dfdxdt =
[          0,        0]
[ -sin(x),           0]
```

【例 5.3.2-2】求 $f=\begin{bmatrix} x_1 e^{x_2} \\ x_2 \\ \cos(x_1)\sin(x_2) \end{bmatrix}$ 的 $jacobian$ 矩阵。

```
syms x1 x2 x3;f=[x1*exp(x2);x2;cos(x1)*sin(x2)];
v=[x1 x2];fjac=jacobian(f,v)
fjac =
[          exp(x2),         x1*exp(x2)]
[                0,                  1]
[ -sin(x1)*sin(x2),   cos(x1)*cos(x2)]
```

5.3.3 符号积分

5.3.3.1 通用积分指令

5.3.3.2 交互式近似积分指令

5.3.3.3 符号积分示例

【例 5.3.3.3-1】求 $\int\begin{bmatrix} ax & bx^2 \\ \frac{1}{x} & \sin x \end{bmatrix}dx$。演示：积分指令对符号函数矩阵的作用。

```
syms a b x;f=[a*x,b*x^2;1/x,sin(x)];
disp('The integral of f is');pretty(int(f))
The integral of f is

                               [        2            3]
                               [1/2 a x      1/3 b x ]
                               [                     ]
                               [ log(x)      -cos(x) ]
```

【例 5.3.3.3-2】求 $\int_0^x \frac{1}{\ln t}dt$ 。演示如何使用 mfun 指令获取一组积分值。

（1）

```
F1=int('1/log(t)','t',0,'x')
F1 =
-Ei(1,-log(x))
```

（2）

```
x=0.5:0.1:0.9
F115=-mfun('Ei',1,-log(x))
x =
    0.5000    0.6000    0.7000    0.8000    0.9000
F115 =
   -0.3787   -0.5469   -0.7809   -1.1340   -1.7758
```

【例 5.3.3.3-3】求积分 $\int_1^2\int_{\sqrt{x}}^{x^2}\int_{\sqrt{xy}}^{x^2y}(x^2+y^2+z^2)dzdydx$。注意：内积分上下限都是函数。

```
syms x y z
F2=int(int(int(x^2+y^2+z^2,z,sqrt(x*y),x^2*y),y,sqrt(x),x^2),x,1,2)
VF2=vpa(F2)
F2 =
```

```
1610027357/6563700-6072064/348075*2^(1/2)+14912/4641*2^(1/4)+64/225*2
^(3/4)
VF2 =
224.92153573331143159790710032805
```

【例 5.3.3.3-4】利用 rsums 求 $S = \int_0^{0.5} \frac{1}{\ln t} dt$ 积分。（与例 5.3.3.3-2 结果比较）

```
syms x positive;px=0.5/log(0.5*x);rsums(px)
```

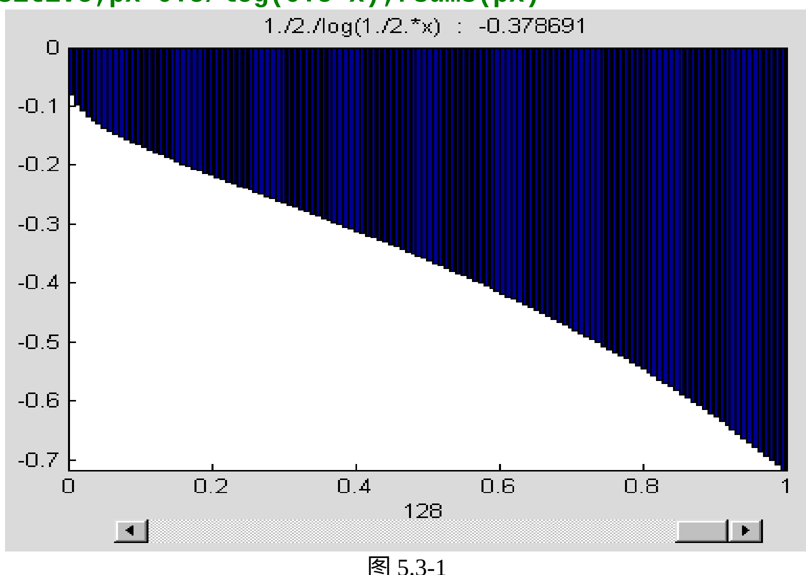


图 5.3-1

5.3.4 符号卷积

【例 5.3.4-1】本例演示卷积的时域积分法：已知系统冲激响应 $h(t) = \frac{1}{T} e^{-t/T} U(t)$，求 $u(t) = e^{-t} U(t)$ 输入下的输出响应。

```
syms T t tao;ut=exp(-t);
ht=exp(-t/T)/T;
uh_tao=subs(ut,t,tao)*subs(ht,t,t-tao);
yt=int(uh_tao,tao,0,t);
yt=simple(yt)
yt =
-(exp(-t)-exp(-t/T))/(T-1)
```

【例 5.3.4-2】本例演示通过变换和反变换求取卷积。系统冲激响应、输入同上例，求输出。

对式(5.3.4-1)两边进行 Laplace 变换得 $L[y(t)] = L[u(t)] * L[h(t)]$，因此有

```
syms s;yt=ilaplace(laplace(ut,t,s)*laplace(ht,t,s),s,t);yt=simple(yt)
yt =
(-exp(-t)+exp(-t/T))/(T-1)
```

【例 5.3.4-3】求函数 $u(t) = U(t) - U(t-1)$ 和 $h(t) = te^{-t} U(t)$ 的卷积。

```
syms tao;t=sym('t','positive');
ut=sym('Heaviside(t)-Heaviside(t-1)');ht=t*exp(-t);
yt=int(subs(ut,t,tao)*subs(ht,t,t-tao),tao,0,t);
yt=collect(yt,'Heaviside(t-1)')
```

```
yt =
(-1+exp(1-t)*t)*Heaviside(t-1)+1+(-t-1)*exp(-t)
```

5.4 符号积分变换

5.4.1 Fourier 变换及其反变换

【例 5.4.1-1】求 $f(t)=\begin{cases}1 & t\ge 0\\ 0 & t<0\end{cases}$ 的 Fourier 变换。本例演示三个重要内容：单位阶跃函数和单位脉冲函数的符号表示；fourier 指令的使用；simple 指令的表现。

（1）求 Fourier 变换

```
syms t w;ut=sym('Heaviside(t)');  %  <1>
UT=fourier(ut)
UTC=maple('convert',UT,'piecewise','w')  %  <3>
UTS=simple(UT)
UT =
pi*Dirac(w)-i/w
UTC =
PIECEWISE([undefined, w = 0],[0, otherwise])
UTS =
-i/w
```

（2）求 Fourier 反变换进行验算

```
Ut=ifourier(UT,w,t)
Uts=ifourier(UTS,w,t)
Ut =
1/2+1/2*Heaviside(t)-1/2*Heaviside(-t)
Uts =
1/2*Heaviside(t)-1/2*Heaviside(-t)
```

【例 5.4.1-2】用 fourier 指令求例 5.1.1-6 中方波脉冲的 Fourier 变换。本例演示：fourier, simple 指令的配合使用。

（1）

```
syms A t w
syms tao positive                           %  <2>
yt=sym('Heaviside(t+tao/2)-Heaviside(t-tao/2)');
Yw=fourier(A*yt,t,w)
Ywc=maple('convert',Yw,'piecewise','w')     %计算结果起指示作用    <5>
Yws=simple(Yw)
Yw =
A*(exp(1/2*i*tao*w)*(pi*Dirac(w)-i/w)-exp(-1/2*i*tao*w)*(pi*Dirac(w)-
i/w))
Ywc =
-i*A*exp(1/2*i*tao*w)/w+i*A*exp(-1/2*i*tao*w)/w
Yws =
2*A*sin(1/2*tao*w)/w
```

（2）

```
Yt=ifourier(Yw,w,t)
Yst=ifourier(Yws,w,t)
Yt =
1/2*A*(Heaviside(t+1/2*tao)-Heaviside(-t-1/2*tao)-Heaviside(t-1/2*tao
)+Heaviside(-t+1/2*tao))
Yst =
```

```
1/2*A*(Heaviside(t+1/2*tao)-Heaviside(-t-1/2*tao)-Heaviside(t-1/2*tao
)+Heaviside(-t+1/2*tao))
```

【例 5.4.1-3】求 $f(t)=\begin{cases} e^{-(t-x)} & t\geq x \\ 0 & t<x \end{cases}$ 的 Fourier 变换，在此 x 是参数，t 是时间变量。

本例演示：fourier 的缺省调用格式的使用要十分谨慎；在被变换函数中包含多个符号变量的情况下，对被变换的自变量给予指明，可保证计算结果的正确。

```
syms t x w;ft=exp(-(t-x))*sym('Heaviside(t-x)');
F1=simple(fourier(ft,t,w))
F2=simple(fourier(ft))
F3=simple(fourier(ft,t))
F1 =
1/exp(i*x*w)/(1+i*w)
F2 =
i*exp(-i*t*w)/(i+w)
F3 =
i*exp(-t*(2+i*t))/(i+t)
```

5.4.2 Laplace 变换及其反变换

【例 5.4.2-1】求 $\begin{bmatrix} \delta(t-a) & u(t-b) \\ e^{-at}\sin bt & t^2\cos 3t \end{bmatrix}$ 的 Laplace 变换。

```
syms t s;syms a b positive                                   %<1>
Dt=sym('Dirac(t-a)');                                        %<2>
Ut=sym('Heaviside(t-b)');                                    %<3>
Mt=[Dt,Ut;exp(-a*t)*sin(b*t),t^2*exp(-t)];MS=laplace(Mt,t,s)
MS =
[        exp(-a*s),       exp(-b*s)/s]
[ b/((s+a)^2+b^2),          2/(1+s)^3]
```

【例 5.4.2-2】验证 Laplace 时移性质：$L\{f(t-t_0)U(t-t_0)\}=e^{-st_0}L\{f(t)\}\quad t_0>0$ 。

```
syms t s;t0=sym('t0','positive');                            %<1>
ft=sym('f(t-t0)')*sym('Heaviside(t-t0)')
FS=laplace(ft,t,s),FS_t=ilaplace(FS,s,t)
ft =
f(t-t0)*Heaviside(t-t0)
FS =
exp(-s*t0)*laplace(f(t),t,s)
FS_t =
f(t-t0)*Heaviside(t-t0)
```

5.4.3 Z 变换及其反变换

【例 5.4.3-1】求序列 $f(n)=\begin{cases} 0 & n<0 \\ 2 & n=0 \\ 6(1-0.5^n) & n>0 \end{cases}$ 的 Z 变换，并用反变换验算。

（1）

```
syms n
Delta=sym('charfcn[0](n)');              % <2>
```

```
D0=subs(Delta,n,0);                    %
D15=subs(Delta,n,15);                  %
disp('[D0,D15]');disp([D0,D15])
[D0,D15]
[ 1, 0]
```

（2）求序列 $f(n)$ 的 Z 变换

```
syms z;fn=2*Delta+6*(1-(1/2)^n);FZ=simple(ztrans(fn,n,z));
disp('FZ = ');pretty(FZ),FZ_n=iztrans(FZ,z,n)
FZ =
                                 2
                              4 z  + 2
                           --------------
                              2
                           2 z  - 3 z + 1
FZ_n =
2*charfcn[0](n)+6-6*(1/2)^n
```

5.5 符号代数方程的求解

5.5.1 线性方程组的符号解

【例 5.5.1-1】求 $d+\frac{n}{2}+\frac{p}{2}=q, n+d+q-p=10, q+d-\frac{n}{4}=p, q+p-n-8d=1$ 线性方程组的解。本例演示，符号线性方程组的基本解法。

该方程组的矩阵形式是 $\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{2} & -1 \\ 1 & 1 & -1 & 1 \\ 1 & -\frac{1}{4} & -1 & 1 \\ -8 & -1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} d \\ n \\ p \\ q \end{bmatrix} = \begin{bmatrix} 0 \\ 10 \\ 0 \\ 1 \end{bmatrix}$。该式简记为 $AX=b$。求符号解的指令如下

```
A=sym([1 1/2 1/2 -1;1 1 -1 1;1 -1/4 -1 1;-8 -1 1 1]);
b=sym([0;10;0;1]);X1=A\b
X1 =
[ 1]
[ 8]
[ 8]
[ 9]
```

【例 5.5.1-2】求解上例前 3 个方程所构成的“欠定”方程组，并解释解的含义。

```
A2=A(1:3,:);X2=A2\b(1:3,1)
syms k;
XX2=X2+k*null(A2)
A2*XX2
X2 =
[ 0]
[ 8]
[ 4]
[ 6]
XX2 =
[   1/3*k]
[       8]
[ 4+4/3*k]
```

```
[      6+k]
ans =
[  0]
[ 10]
[  0]
```

5.5.2 一般代数方程组的解

【例 5.5.2-1】求方程组 $uy^\wedge 2+vz+w=0$， $y+z+w=0$ 关于 y,z 的解。

```
S=solve('u*y^2+v*z+w=0','y+z+w=0','y','z')
disp('S.y'),disp(S.y),disp('S.z'),disp(S.z)
S =
    y: [2x1 sym]
    z: [2x1 sym]
S.y
[ -1/2/u*(-2*u*w-v+(4*u*w*v+v^2-4*u*w)^(1/2))-w]
[ -1/2/u*(-2*u*w-v-(4*u*w*v+v^2-4*u*w)^(1/2))-w]
S.z
[ 1/2/u*(-2*u*w-v+(4*u*w*v+v^2-4*u*w)^(1/2))]
[ 1/2/u*(-2*u*w-v-(4*u*w*v+v^2-4*u*w)^(1/2))]
```

【例 5.5.2-2】用 solve 指令重做例 5.5.1-2。即求 $d+\frac{n}{2}+\frac{p}{2}=q$， $n+d+q-p=10$， $q+d-\frac{n}{4}=p$ 构成的“欠定”方程组解。

```
syms d n p q;eq1=d+n/2+p/2-q;eq2=n+d+q-p-10;eq3=q+d-n/4-p;
S=solve(eq1,eq2,eq3,d,n,p,q);S.d,S.n,S.p,S.q
Warning: 3 equations in 4 variables.
> In D:\MATLAB6P5\toolbox\symbolic\solve.m at line 110
  In D:\MATLAB6P5\toolbox\symbolic\@sym\solve.m at line 49
ans =
d
ans =
8
ans =
4*d+4
ans =
3*d+6
```

【例 5.5.2-3】求 $(x+2)^x=2$ 的解。

```
clear all,syms x;s=solve('(x+2)^x=2','x')
s =
.69829942170241042826920133106081
```

5.6 符号微分方程的求解

5.6.1 符号解法和数值解法的互补作用

5.6.2 求微分方程符号解的一般指令

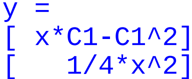
5.6.3 微分方程符号解示例

【例 5.6.3-1】求 $\dfrac{dx}{dt}=y,\ \dfrac{dy}{dt}=-x$ 的解。

```
S=dsolve('Dx=y,Dy=-x');
disp([blanks(12),'x',blanks(21),'y']),disp([S.x,S.y])
            x                      y
[  cos(t)*C1+sin(t)*C2, -sin(t)*C1+cos(t)*C2]
```

【例 5.6.3-2】图示微分方程 $y=xy'-(y')^2$ 的通解和奇解的关系。

```
y=dsolve('y=x*Dy-(Dy)^2','x')
clf,hold on,ezplot(y(2),[-6,6,-4,8],1)
cc=get(gca,'Children');                          % <3>
set(cc,'Color','r','LineWidth',5)                % <4>
for k=-2:0.5:2;ezplot(subs(y(1),'C1',k),[-6,6,-4,8],1);end
hold off,title('\fontname{隶书}\fontsize{16}通解和奇解')
y =
```



```
[ x*C1-C1^2]
[   1/4*x^2]
```

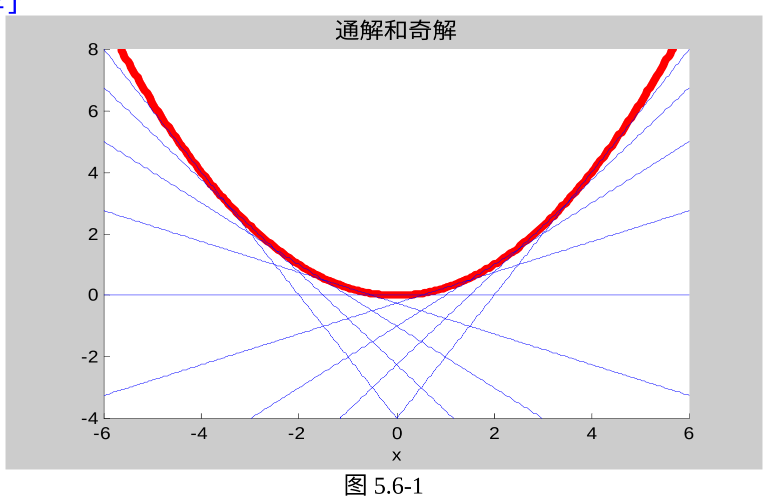


图 5.6-1

【例 5.6.3-3】求解两点边值问题：$xy''-3y'=x^2, y(1)=0, y(5)=0$ 。（注意：相应的数值解法比较复杂）。

```
y=dsolve('x*D2y-3*Dy=x^2','y(1)=0,y(5)=0','x')
y =
-1/3*x^3+125/468+31/468*x^4
```

【例 5.6.3-4】求边值问题 $\dfrac{df}{dx}=3f+4g,\ \dfrac{dg}{dx}=-4f+3g,\ \ f(0)=0,\ \ g(0)=1$ 的解。（注意：相应的数值解法比较复杂）。

```
S=dsolve('Df=3*f+4*g,Dg=-4*f+3*g','f(0)=0,f(3)=1')
S.f,S.g
S =
    f: [1x1 sym]
    g: [1x1 sym]
ans =
exp(3*t)*sin(4*t)/sin(12)/(cosh(9)+sinh(9))
ans =
exp(3*t)*cos(4*t)/sin(12)/(cosh(9)+sinh(9))
```

5.7 利用 MAPLE 的深层符号计算资源

5.7.1 经典特殊函数的调用

5.7.2 MAPLE 库函数在线帮助的检索树

5.7.3 发挥 MAPLE 的计算潜力

5.7.3.1 调用 MAPLE 函数

【例 5.7.3.1-1】求递推方程 $f(n) = -3f(n-1) - 2f(n-2)$ 的通解。

（1）

```
gs1=maple('rsolve(f(n)=-3*f(n-1)-2*f(n-2),f(k));')
gs1 =
(2*f(0)+f(1))*(-1)^k+(-f(0)-f(1))*(-2)^k
```

（2）调用格式二

```
gs2=maple('rsolve','f(n)=-3*f(n-1)-2*f(n-2)','f(k)')
gs2 =
(2*f(0)+f(1))*(-1)^k+(-f(0)-f(1))*(-2)^k
```

【例 5.7.3.1-2】求 $f = xyz$ 的 Hessian 矩阵。

（1）

```
FH1=maple('hessian(x*y*z,[x,y,z]);')
FH1 =
matrix([[0, z, y], [z, 0, x], [y, x, 0]])
```

（2）

```
FH2=maple('hessian','x*y*z','[x,y,z]')
FH2 =
matrix([[0, z, y], [z, 0, x], [y, x, 0]])
```

（3）

```
FH=sym(FH2)
FH =
[ 0, z, y]
[ z, 0, x]
[ y, x, 0]
```

【例 5.7.3.1-3】求 $\sin(x^2 + y^2)$ 在 $x = 0, y = 0$ 处展开的截断 8 阶小量的台劳近似式。

（1）

```
TL1=maple('mtaylor(sin(x^2+y^2),[x=0,y=0],8)')
TL1 =
mtaylor(sin(x^2+y^2),[x = 0, y = 0],8)
```

（2）

```
maple('readlib(mtaylor);');
TL2=maple('mtaylor(sin(x^2+y^2),[x=0,y=0],8)');
pretty(sym(TL2))
```

```
   2     2          6         2    4         4    2         6
x~  + y   - 1/6 x~   - 1/2 y   x~   - 1/2 y   x~   - 1/6 y
```

5.7.3.2 运行 MAPLE 程序

【例 5.7.3.2-1】目标：设计求取一般隐函数 $f(x,y)=0$ 的导数 $y'(x)$ 解析解的程序，并要求：该程序能象 MAPLE 原有函数一样可被永久调用。

（1）

[DYDZZY.src]

```
DYDZZY:=proc(f)
# DYDZZY(f) is used to get the derivate of
#       an implicit function
local Eq,deq,imderiv;
Eq:='Eq';
Eq:=f;
deq:=diff(Eq,x);
readlib(isolate);
imderiv:=isolate(deq,diff(y(x),x));
end;
```

（2）

```
procread('DYDZZY.src')
ans =
DYDZZY := proc (f) local Eq, deq, imderiv; Eq := 'Eq'; Eq := f; deq :=
diff(Eq,x); readlib(isolate); imderiv := isolate(deq,diff(y(x),x)) end
```

（3）

```
s1=maple('DYDZZY(x=log(x+y(x));')
s2=maple('DYDZZY(x^2*y(x)-exp(2*x)=sin(y(x)))')
s3=maple('DYDZZY','cos(x+sin(y(x)))=sin(y(x))')
s1 =
diff(y(x),x) = x+y(x)-1
s2 =
diff(y(x),x) = (-2*x*y(x)+2*exp(2*x))/(x^2-cos(y(x)))
s3 =
diff(y(x),x)                                                  =
sin(x+sin(y(x)))/(-sin(x+sin(y(x)))*cos(y(x))-cos(y(x)))
```

（4）

```
clear maplemex
procread('DYDZZY.src');
maple('save(`DYDZZY.m`)');
```

（5）

```
maple('read','`DYDZZY.m`');
ss2=maple('DYDZZY(x^2*y(x)-exp(2*x)=sin(y(x)))')
ss2 =
diff(y(x),x) = (-2*x*y(x)+2*exp(2*x))/(x^2-cos(y(x)))
```

5.7.3.3 数值、符号计算集成 M 文件的编写

5.8 可视化数学分析界面

5.8.1 单变量函数分析的交互界面

`funtool`

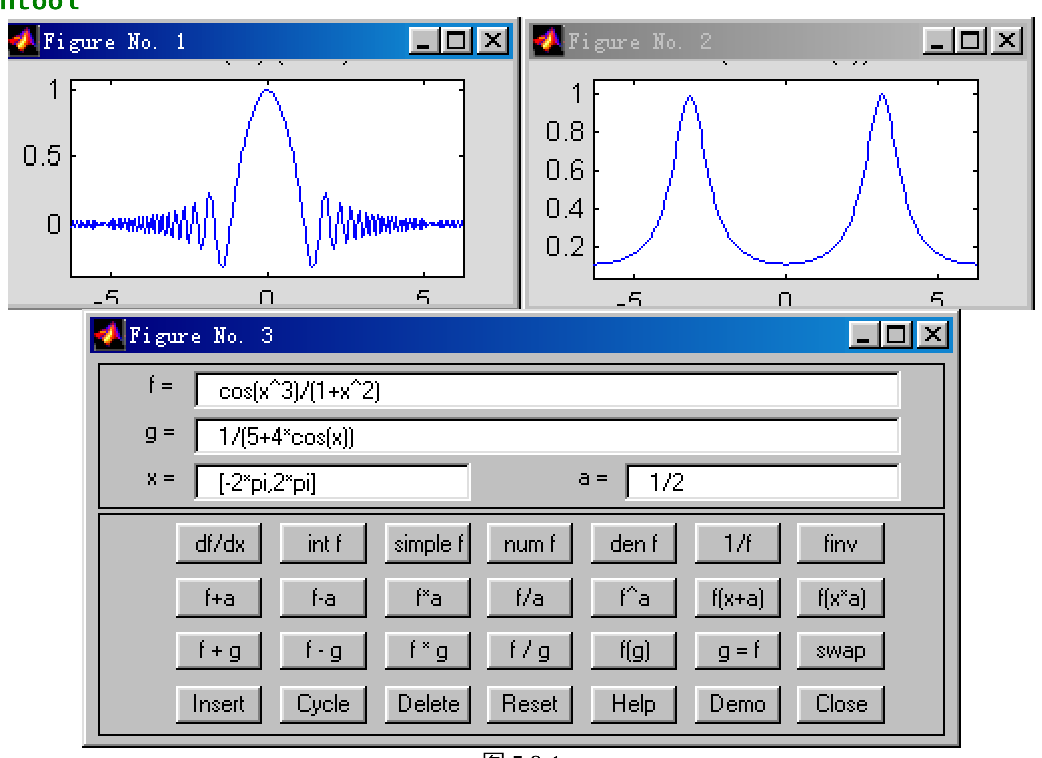


图 5.8-1

5.8.2 泰勒级数逼近分析界面

`taylortool`

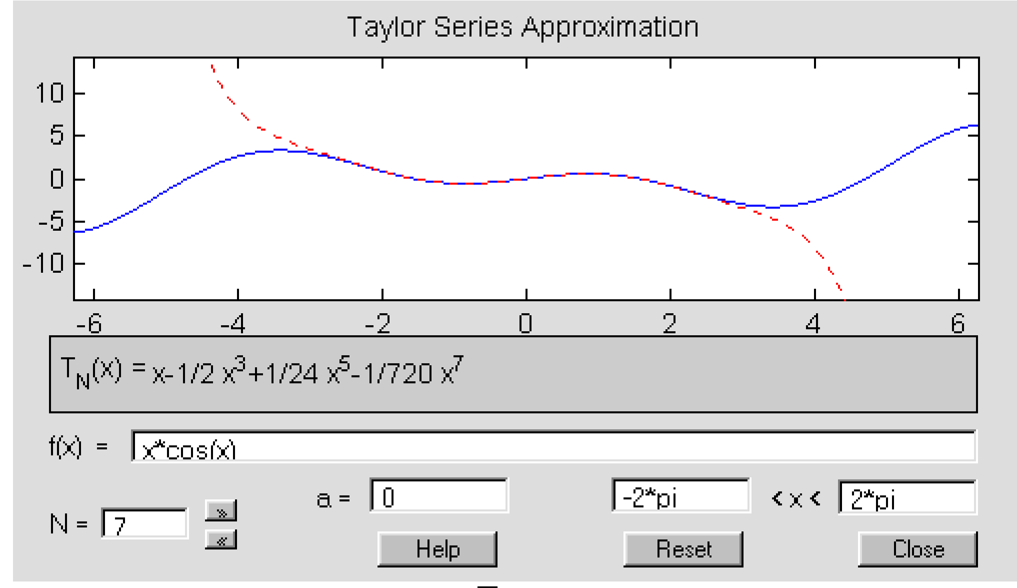


图 5.8-2

第七章　M 文件和面向对象编程

假如读者想灵活运用 MATLAB 去解决实际问题，想充分调动 MATLAB——科学技术资源，想理解 MATLAB 版本升级所依仗的基础，那么本章内容将十分有用。

本章将涉及比较深层的 MATLAB 内容：脚本；函数（一般函数、内联函数、子函数、私用函数、方法函数）；函数句柄的创建和使用；程序调试和剖析；数据结构（类、对象）；重载和继承；面向对象编程。本章配备了许多精心设计的算例。这些算例是完整的，可直接演练的。读者通过这些算例，将真切感受到抽象概念的内涵、各指令间的协调，将从感知上领悟到面向对象编程的优越和至关要领。

本章新增了第 7.7 节，专门阐述函数句柄的创建和使用，它适用于 MATLAB6.x 版；而新增的第 7.9.3 节中关于程序性能优化的内容，则仅适用于 MATLAB6.5 以后版。

7.1　入门

【例 7.1-1】通过 M 脚本文件，画出下列分段函数所表示的曲面。

$$p(x_1,x_2)=\begin{cases}0.5457e^{-0.75x_2^2-3.75x_1^2-1.5x_1} & x_1+x_2>1\\ 0.7575e^{-x_2^2-6x_1^2} & -1<x_1+x_2\le 1\\ 0.5457e^{-0.75x_2^2-3.75x_1^2+1.5x_1} & x_1+x_2\le -1\end{cases}$$

（1）

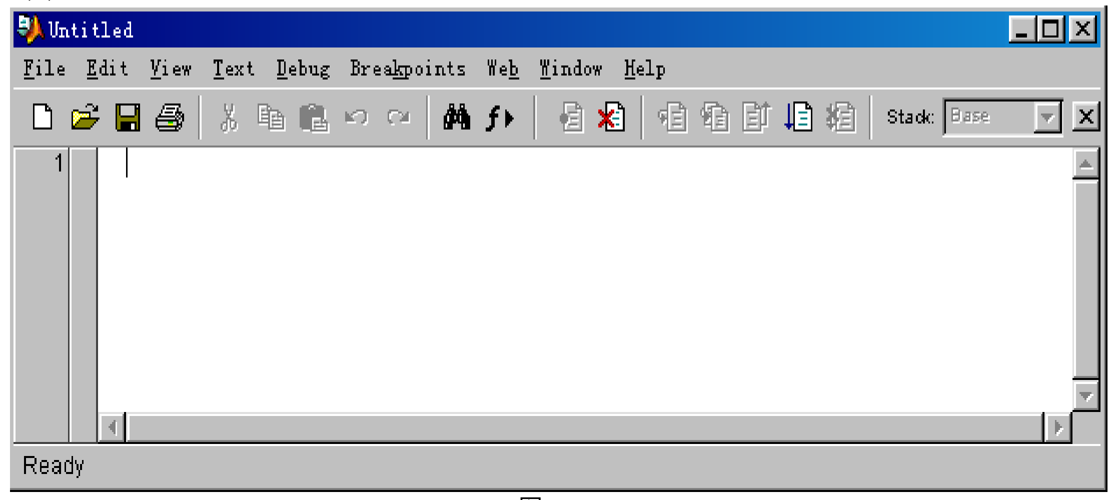


图 7.1-1

[exm0701_1.m]

```
%exm0701_1.m
a=2;b=2;                                                    %         <2>
clf;
x=-a:0.2:a;y=-b:0.2:b;
for i=1:length(y)
   for j=1:length(x)
      if x(j)+y(i)>1
         z(i,j)=0.5457*exp(-0.75*y(i)^2-3.75*x(j)^2-1.5*x(j));
      elseif x(j)+y(i)<=-1
         z(i,j)=0.5457*exp(-0.75*y(i)^2-3.75*x(j)^2+1.5*x(j));
```

```
        else z(i,j)=0.7575*exp(-y(i)^2-6.*x(j)^2);
        end
    end
end
axis([-a,a,-b,b,min(min(z)),max(max(z))]);
colormap(flipud(winter));surf(x,y,z);
```

（2）

```
exm0701_1
```

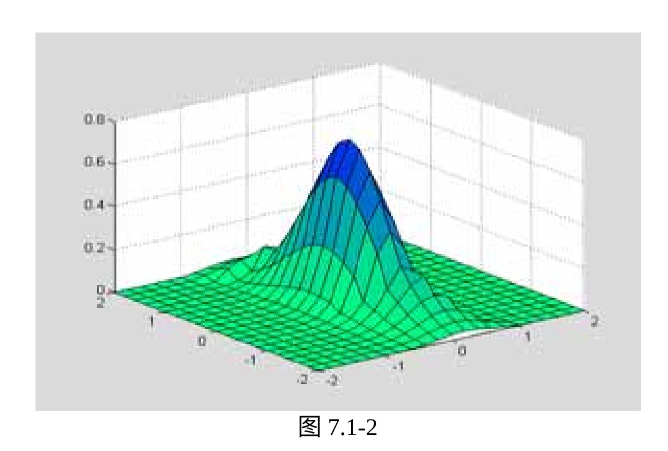


图 7.1-2

【例 7.1-2】通过 M 函数文件画出上例分段函数的曲面。

```
exm0701_2(2,2)
```

7.2 M 文本编辑器

7.3 MATLAB 控制流

7.3.1 for 循环结构

【例 7.3.1-1】一个简单的 for 循环示例。

```
for i=1:10;
      x(i)=i;
end;
x
x =
     1     2     3     4     5     6     7     8     9    10
```

7.3.2 while 循环结构

【例 7.3.2-1】Fibonacci 数组的元素满足 Fibonacci 规则：$a_{k+2}=a_k+a_{k+1}$，$(k=1,2,\Lambda)$；且 $a_1=a_2=1$。现要求该数组中第一个大于 10000 的元素。

```
a(1)=1;a(2)=1;i=2;
while  a(i)<=10000
        a(i+1)=a(i-1)+a(i);
        i=i+1;
```

```
end;
i,a(i),
i =
    21
ans =
       10946
```

7.3.3 if-else-end 分支结构

【例 7.3.3-1】一个简单的分支结构。

```
cost=10;number=12;
if number>8
   sums=number*0.95*cost;
end,sums
sums =
  114.0000
```

【例 7.3.3-2】用 for 循环指令来寻求 Fibonacc 数组中第一个大于 10000 的元素。

```
n=100;a=ones(1,n);
for i=3:n
     a(i)=a(i-1)+a(i-2);
     if a(i)>=10000
        a(i),
        break;
     end;
end,i
ans =
       10946
i =
    21
```

7.3.4 switch-case 结构

【例 7.3.4-1】学生的成绩管理，用来演示 switch 结构的应用。

```
clear;
%
for i=1:10;a{i}=89+i;b{i}=79+i;c{i}=69+i;d{i}=59+i;end;c=[d,c];
Name={' Jack','Marry','Peter',' Rose','  Tom'};
Mark={72,83,56,94,100};Rank=cell(1,5);
%
S=struct('Name',Name,'Marks',Mark,'Rank',Rank);
%
for i=1:5
   switch S(i).Marks
   case 100
      S(i).Rank='满分';
   case a
      S(i).Rank=' 优秀';
   case b
      S(i).Rank=' 良好';
   case c
      S(i).Rank=' 及格';
```

```
   otherwise
      S(i).Rank='不及格';
   end
end
%
disp(['学生姓名  ','  得分  ','    等级']);disp(' ')
for i=1:5;
 disp([S(i).Name,blanks(6),num2str(S(i).Marks),blanks(6),S(i).Rank]);
end;
学生姓名     得分       等级

 Jack      72        及格
Marry      83        良好
Peter      56       不及格
 Rose      94        优秀
  Tom      100       满分
```

7.3.5 try-catch 结构

【例 7.3.5-1】try-catch 结构应用实例：对 (3×3) 魔方阵的行进行援引，当"行下标"超出魔方阵的最大行数时，将改向对最后一行的援引，并显示"出错"警告。

```
clear,N=4;A=magic(3);
try
     A_N=A(N,:)
catch
     A_end=A(end,:)
end
lasterr
A_end =
     4     9     2
ans =
 Index exceeds matrix dimensions.
```

7.3.6 控制程序流的其它常用指令

7.3.6.1 return 指令

7.3.6.2 input 和 keyboard 指令

7.3.6.3 yesinput 指令

7.3.6.4 pause 指令

7.3.6.5 break 指令

7.3.6.6 error 和 warning 指令

7.4 脚本文件和函数文件

7.4.1 M 脚本文件

7.4.2 M 函数文件

7.4.3 局部变量和全局变量

7.4.4 M 文件的一般结构

【例 7.4.4-1】M 函数文件示例。本例演示：（A）编写一个画任意半径任意色彩线型的圆。（B）完整函数文件的基本结构。（C）函数文件各基本组成部分的作用。

[exm07044_1.m]

```
function sa = exm07044_1(r,s)
%CIRCLE
%
%

%
if nargin>2
   error('输入宗量太多。');
end;
if nargin==1
   s='b';
end;
clf;
t=0:pi/100:2*pi;
x=r*exp(i*t);
if nargout==0
   plot(x,s);
else
   sa=pi*r*r;
   fill(real(x),imag(x),s)
end
axis('square')
```

7.4.5 P 码文件

7.4.5.1 语法分析过程和伪代码

7.4.5.2 P 码文件的预生成

7.4.5.3 内存中 P 码文件的列表和清除

7.4.6 MATLAB 的搜索过程

7.5 变量的检测传递和限权使用函数

7.5.1 输入输出宗量检测指令

7.5.2 “ 变长度 ” 输入输出宗量

【例 7.5.2-1】变长度宗量使用示例。

（1）

[exm07052_1.m]

```
function varargout = exm07052_1(r,varargin)
%RINGZY      Plot a ring and calculate the area of the ring.
%
%
vin=length(varargin);Nin=vin+1;          %   <11>
error(nargchk(1,Nin,nargin))             %
if nargout>6                             %
   error('Too many output arguments')
end
t=0:pi/20:2*pi;x=r*exp(i*t);s=pi*r*r;
if nargout==0
   switch Nin
   case 1
      plot(x,'b')
   case 2
      r2=varargin{1};                                              %<22>
      x2=r2*exp(i*t);
      plot(x,'b');hold on ;plot(x2,'b');hold off
   otherwise
      r2=varargin{1};                                              %<26>
      x2=r2*exp(i*t);
      plot(x,varargin{2:end});hold on       %     <28>
      plot(x2,varargin{2:end});hold off     %     <29>
   end;
   axis('square')
else
   varargout{1}=real(x);varargout{2}=imag(x);                      %<33>
   varargout{5}=pi*r*r;varargout{6}=[];                            %<34>
   if Nin>1
      r2=varargin{1};                                              %<36>
      x2=r2*exp(i*t);
      varargout{3}=real(x2);varargout{4}=imag(x2);                 %<38>
      varargout{6}=pi*(r^2-r2^2);                                  %<39>
   end;
end
```

（2）

```
r1=1;r2=3;
[x1,y1,x2,y2,s1,s2]=exm07052_1(r1);
[x1,y1,x2,y2]=exm07052_1(r1,r2);
[x1,y1,x2,y2,s1,s2]=exm07052_1(r1,r2);
```

（3）

```
r1=1;r2=0.6;
subplot(1,3,1),exm07052_1(r1,r2),
```

```
subplot(1,3,2),exm07052_1(r1,r2,'Marker','o')
subplot(1,3,3),exm07052_1(r1,r2,'LineWidth',5,'Color',[1 0.4 0])
```

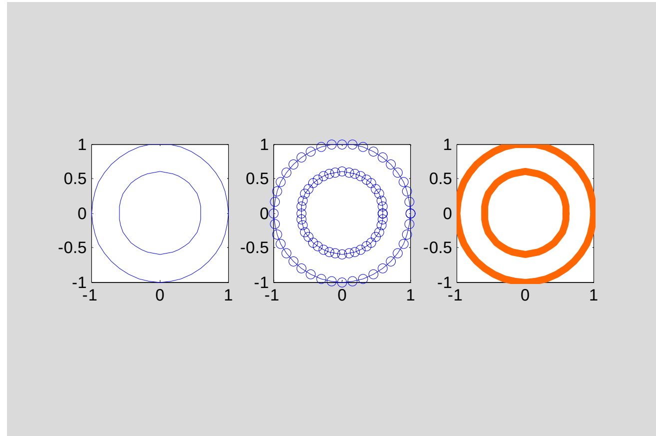


图 7.5-1

7.5.3 跨空间变量传递

7.5.3.1 跨空间计算串表达式的值

【例 7.5.3.1-1】本例演示：（A）编写绘制正多边形或圆的程序。（B）子函数与（母）函数的关系。（C）各种不同的工作空间。（D）evalin 运行机理与 eval 的异同。

（1）

[exm070531_1.m]

```
function y1=exm070531_1(a,s)
t=(0:a)/a*2*pi;
y1=subevalinzzy(4,s);
%------------ subfunction -------------
function y2=subevalinzzy(a,s)
t=(0:a)/a*2*pi;ss='a*exp(i*t)';
switch s
case {'base','caller'}
   y2=evalin(s,ss);
case 'self'
   y2=eval(ss);
end
```

（2）

```
clear,a=30;t=(0:a)/a*2*pi;sss={'base','caller','self'};
for k=1:3
   y0=exm070531_1(8,sss{k});
   subplot(1,3,k)
   plot(real(y0),imag(y0),'r','LineWidth',3),axis square image
end
```

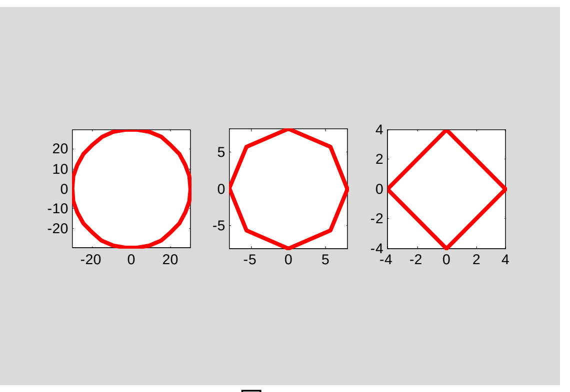


图 7.5-2

7.5.3.2 跨空间赋值

【例 7.5.3.2-1】assignin 运作机理示范。

（1）

[exm070532_1.m]

```
function y=exm070532_1(x)
y=sqrt(x);t=x^2;
assignin('base','yy',t)
```

（2）

```
clear;x=4;y=exm070532_1(x);
disp([blanks(5),'x',blanks(5),'y',blanks(4),'yy']),disp([x,y,yy])
     x     y    yy
     4     2    16
```

7.5.4 子函数和私用函数

7.5.4.1 子函数

7.5.4.2 私用函数

7.6 串演算函数

7.6.1 eval

【例 7.6.1-1】计算 “ 表达式 ” 串，产生向量值。

```
clear,t=pi;cem='[t/2,t*2,sin(t)]';y=eval(cem)
y =
    1.5708    6.2832    0.0000
```

【例 7.6.1-2】计算 “ 语句 ” 串，创建变量。

```
clear,t=pi;eval('theta=t/2,y=sin(theta)');who
theta =
    1.5708
y =
     1
```

```
Your variables are:

t       theta   y
```

【例 7.6.1-3】计算 “ 替代 ” 串。

```
A=ones(2,1);B=ones(1,3);c=eval('B*A','A*B'),errmessage=lasterr
c =
     1     1     1
     1     1     1
errmessage =
Error using ==> *
Inner matrix dimensions must agree.
```

【例 7.6.1-4】计算 “ 合成 ” 串。

```
CEM={'cos','sin','tan'};
for k=1:3
    theta=pi*k/12;
    y(1,k)=eval([CEM{1},'(',num2str(theta),')']);
end
y
y =
    0.9659    0.8660    0.7071
```

7.6.2 feval

【例 7.6.2-1】feval 和 eval 运行区别之一：feval 的 FN 绝对不能是表达式。

```
x=pi/4;Ve=eval('1+sin(x)')
Ve =
    1.7071

Vf=feval('1+sin(x)',x)
??? Error using ==> feval
Invalid function name '1+sin(x)'.
```

【例 7.6.2-2】feval 和 eval 调用区别：feval 的 FN 只接受函数名。本例两种方法以后者为好。

```
randn('seed',1);A=rand(2,2);
[ue,de,ve]=eval('svd(A)');
disp('Results by eval');disp([ue,de,ve]);disp(blanks(1))
[uf,df,vf]=feval('svd',A);
disp('Results by feval');disp([uf,df,vf])
Results by eval
   -0.9193   -0.3936    1.2212         0   -0.7897   -0.6135
   -0.3936    0.9193         0    0.2633   -0.6135    0.7897

Results by feval
   -0.9193   -0.3936    1.2212         0   -0.7897   -0.6135
   -0.3936    0.9193         0    0.2633   -0.6135    0.7897
```

7.6.3 内联函数

7.6.3.1 内联函数的创建

7.6.3.2 涉及内联函数性质的指令

7.6.3.3 内联函数创建和应用示例

【例 7.6.3.3-1】演示：内联函数的第一种创建格式；使内联函数适于“ 数组运算 ”。

```
clear,F1=inline('sin(rho)/rho')
F1 =
     Inline function:
     F1(rho) = sin(rho)/rho
f1=F1(2)
f1 =
    0.4546
FF1=vectorize(F1)
xx=[0.5,1,1.5,2];ff1=FF1(xx)
FF1 =
     Inline function:
     FF1(rho) = sin(rho)./rho
ff1 =
    0.9589    0.8415    0.6650    0.4546
```

【例 7.6.3.3-2】演示：第一种内联函数创建格式的缺陷；含向量的多宗量输入的赋值。

```
G1=inline('a*exp(x(1))*cos(x(2))'),G1(2,[-1,pi/3])
G1 =
     Inline function:
     G1(a) = a*exp(x(1))*cos(x(2))
??? Error using ==> inline/subsref
Too many inputs to inline function.
G2=inline('a*exp(x(1))*cos(x(2))','a','x'),G2(2,[-1,pi/3])
G2 =
     Inline function:
     G2(a,x) = a*exp(x(1))*cos(x(2))
ans =
    0.3679
```

【例 7.6.3.3-3】演示：产生向量输入、向量输出的内联函数；这种向量函数的调用方法。

```
Y2=inline('[x(1)^2;3*x(1)*sin(x(2))]')
argnames(Y2)
Y2 =
     Inline function:
     Y2(x) = [x(1)^2;3*x(1)*sin(x(2))]
ans =
    'x'
x=[4,pi/6];
y2=Y2(x)
y2 =
   16.0000
    6.0000
```

【例 7.6.3.3-4】演示：最简练格式创建内联函数；内联函数可被 feval 指令调用。

```
Z2=inline('P1*x*sin(x^2+P2)',2)
Z2 =
     Inline function:
     Z2(x,P1,P2) = P1*x*sin(x^2+P2)
z2=Z2(2,2,3)
fz2=feval(Z2,2,2,3)
z2 =
    2.6279
fz2 =
```

```
    2.6279
```

7.7 函数句柄

7.7.1 函数句柄的创建和观察

【例 7.7.1-1】为 MATLAB 的 “ 内建 ” 函数创建函数句柄，并观察其内涵。

（1）

```
hsin=@sin;
```

（2）

```
class(hsin)
size(hsin)
ans =
function_handle
ans =
     1     1
```

（3）

```
CC=functions(hsin)
CC =
    function: 'sin'
        type: 'overloaded'
        file: 'MATLAB built-in function'
     methods: [1x1 struct]
```

（4）

```
CC.methods.sym
ans =
d:\matlab6p5\toolbox\symbolic\@sym\sin
```

7.7.2 函数句柄的基本用法

【例 7.7.2-1】本例通过函数及其句柄演示若干基本用法。

（1）

```
fhandle=str2func('sin');
```

（2）

```
ys=sin(pi/4)
yfold=feval('sin',pi/4)
yfnew=feval(fhandle,pi/4)
ys =
    0.7071
yfold =
    0.7071
yfnew =
    0.7071
```

（3）

```
Alpha=sym('pi/4');
yss=sin(Alpha)
yfold=feval('sin',Alpha)
ynews=feval('sin',Alpha)
yss =
```

```
1/2*2^(1/2)
yfold =
1/2*2^(1/2)
ynews =
1/2*2^(1/2)
```

（4）

```
xold=fminbnd('sin',0,2*pi)
xnew=fminbnd(fhandle,0,2*pi)
xold =
    4.7124
xnew =
    4.7124
```

【例 7.7.2-2】本例演示：如何避免创建“ 无效函数句柄“ 问题。

（1）

```
Hy2=@fhzzy          %fhzzy.m 是随书光盘 mfiles 文件夹上的一个函数文件。
Hy2 =
    @fhzzy
```

（2）

```
class(Hy2)
size(Hy2)
ans =
function_handle
ans =
     1     1
```

（3）

```
feval(Hy2,'line');
??? Error using ==> feval
Undefined function 'fhzzy'.
```

【例 7.7.2-3】自建函数及其句柄的使用。

（1）

[fhzzy.m]

```
function Hr=fhzzy(flag )
% fhzzy
%
%

t=(0:100)/100*2*pi;
x=sin(t);
y=cos(t);
Hr=@cirline;
feval(Hr,flag,x,y,t)
% -------------subfunction---------------------------
function cirline(wd,x,y,t)
%
%
switch wd
case 'line'
   plot(t,x,'b' ,t,y,'r','LineWidth',2)
case 'circle'
   plot(x,y,'g.','MarkerSize',30),
   axis square off
```

```
otherwise
   error('输入宗量只能取 "line" 或 "circle" ！ ')
end
shg
```

（2）

```
Hy3=@fhzzy
fhzzy('line');
Hy3 =
    @fhzzy
```

图 7.7-1

（3）

```
which('fhzzy')
fhzzy not found.
```

（4）

```
fhzzy('line')
feval('fhzzy','line')
??? Undefined function or variable 'fhzzy'.
```

（5）

```
feval(Hy3,'line');
```

【例 7.7.2-4】子函数句柄的创建与使用。

（1）

```
HCL=fhzzy('circle')
HCL =
    @cirline
```

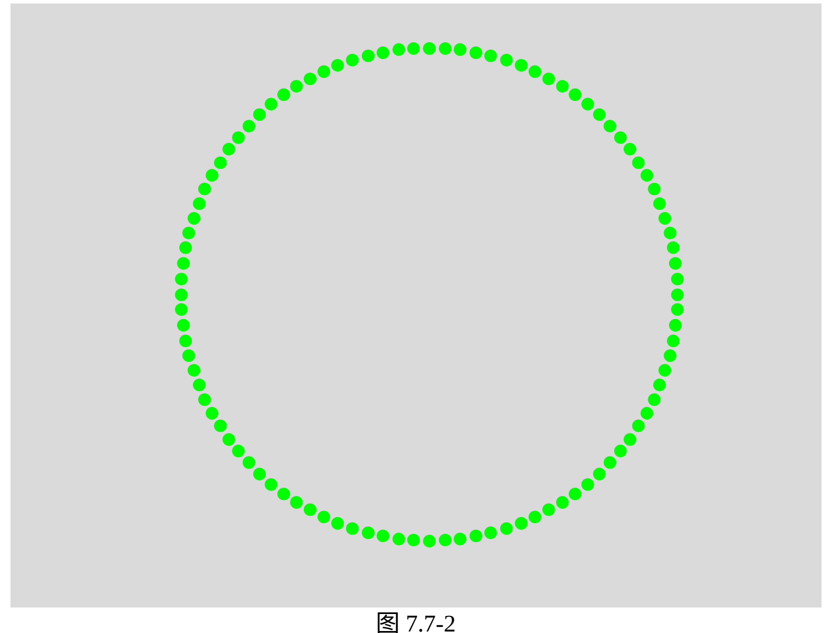


图 7.7-2

（2）

```
tt=(0:100)/100*2*pi;
xx=sin(tt);
yy=cos(tt);
cirline('circle',xx,yy,tt);
feval('circle',xx,yy,tt)
??? Undefined function or variable 'cirline'.
```

（3）

```
feval(HCL,'circle',xx,yy,tt)
```

7.8 创建用户工具箱

7.8.1 MATLAB 对工具箱文件的管理特点

7.8.2 建立用户工具箱须知

7.9 调试和剖析

7.9.1 直接调试法

7.9.2 调试器的使用

7.9.2.1 图形式调试器

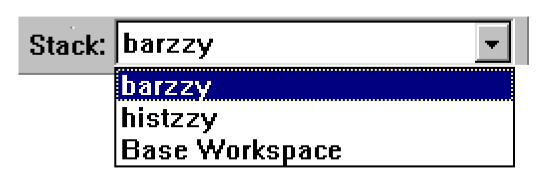


图 7.9-1

7.9.2.2 调试器应用示例

【例 7.9.2.2.-1】本例的目标：对于任意随机向量，画出鲜明标志该随机向量均值、标准差的频数直方图（如图 7.9-2），或给出绘制这种图形的数据。

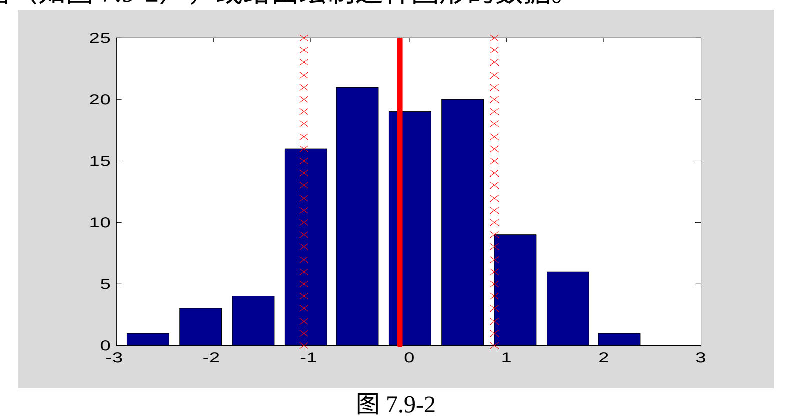


图 7.9-2

（1）

[exm070922_1.m]

```
function [nn,xx,xmu,xstd]=exm070922_1(x)
%
xmu=mean(x);
xstd=std(x);
[nn,xx]=hist(x);
if nargout==0
   barzzy0(nn,xx,xmu,xstd)                          %<7>
end
```

[barzzy0.m]

```
function barzzy0(nn,xx,xmu,xstd)
%
%
clf,
bar(xx,nn);hold on
Ylimit=get(gca,'YLim');
yy=0:Ylimit(2);
xxmu=xmu*size(yy);
xxL=xxmu/xmu*(xmu-xstd);
xxR=xxmu/xmu*(xmu+xstd);
plot(xxmu,yy,'r','Linewidth',3)                     %<11>
plot(xxL,yy,'rx','MarkerSize',8)
plot(xxR,yy,'rx','MarkerSize',8),hold off
```

（2）

```
randn('seed',1),x=randn(1,100);exm070922_1(x);
??? Error using ==> plot
```

```
Vectors must be the same lengths.

Error in ==> D:\Master6\mfile\barzzy0.m
On line 11  ==> plot(xxmu,yy,'r','Linewidth',3)   %<11>

Error in ==> D:\Master6\mfile\exm070922_1.m
On line 7  ==>    barzzy0(nn,xx,xmu,xstd)         %<7>
```

图 7.9-3

(3)

(4)

(5)

```
randn('seed',1),x=randn(1,100);exm070922_1(x);
```

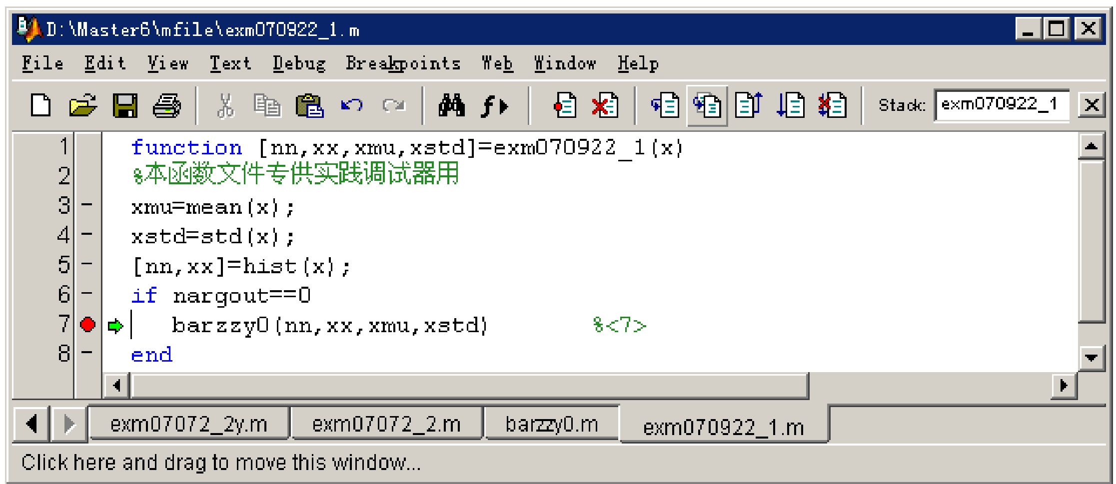


图 7.9-3

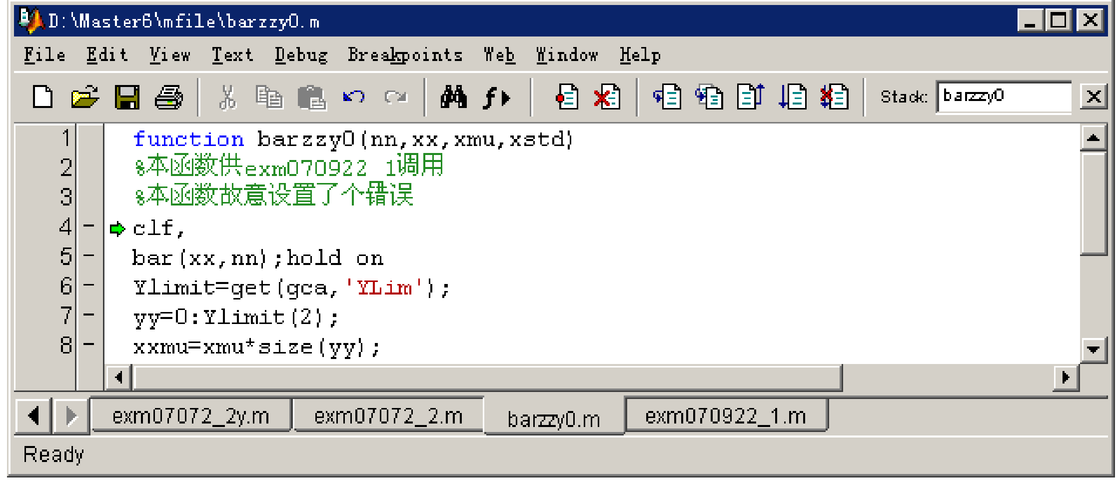


图 7.9-4

（7）

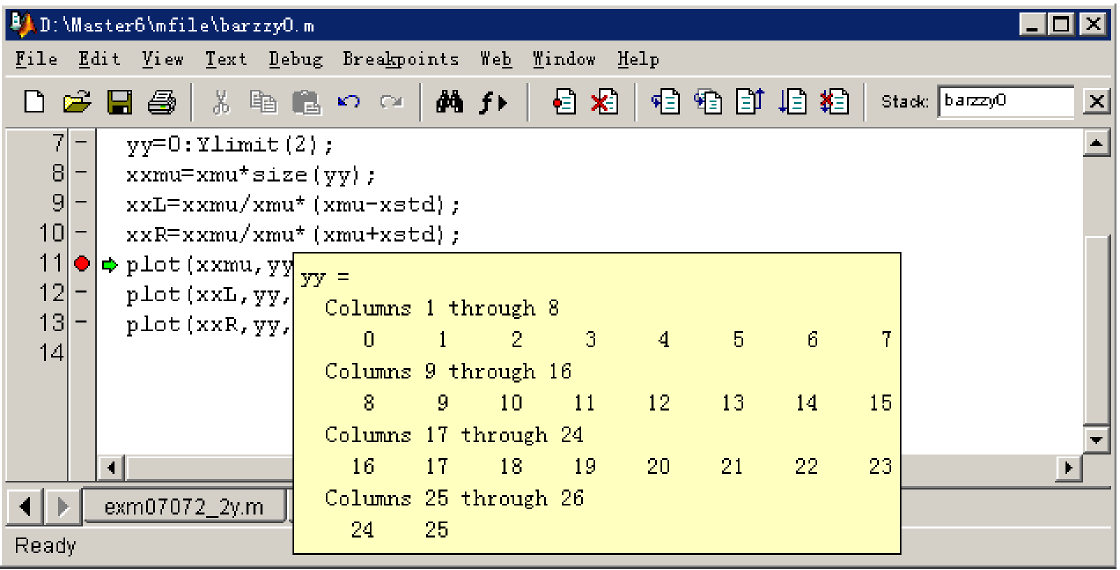


图 7.9-5

（9）

```
randn('seed',1),x=randn(1,100);exm070922_1(x);
```

7.9.3 MATLAB 程序的性能优化

7.9.3.1 提高 MATLAB 运行速度的有效措施

7.9.3.2 JIT 和加速器的加速能力

【例 7.9.3.2-1】试验 JIT 和加速器对 M 文件的加速作用。

7.9.3.3 程序性能的剖析

【例 7.9.3.3-1】演示界面式剖析器的使用。本例被剖析文件 relaxzzy.m 可从随书光盘的 mfiles 目录上得到。

（1）



图 7.9-6

（2）

方法一：

方法二：

（3）

Lines where the most time was spent [table | list | hide]

Line Number	Code	Calls	Total Time	% Time	Time Plot
27	drawnow	60	1.740 s	26.6%	
19	newplate(j,k)=(plate(jm1,...	3000000	1.537 s	23.5%	
31	close	1	0.280 s	4.3%	
6	im=image(plate);	1	0.280 s	4.3%	
22	end	3000000	0.116 s	1.8%	
All other lines			2.587 s	39.6%	
Totals			6.540 s	100%	

图 7.9-7

File listing

Color highlight code according to [Time | Number of Calls | Coverage | Acceleration | No Color]

time	calls	acc	line	
			1	function elapt=relaxzzy(iter)
			2	% 本程序供试验 JIT 和加速器的作用
0.000	1	.	3	sz=102;
0.050	1	x	4	plate=magic(sz)*64/(sz*sz);
0.000	1	.	5	newplate=plate;
0.280	1	x	6	im=image(plate);
0.050	1	x	7	axis off
	1	x	8	set(gcf,'DoubleBuffer','on')
			9	
	1	x	10	shg
0.000	1	.	11	tic
	1	x	12	for i=1:iter
0.001	300	.	13	for j=2:(sz-1)
0.001	30000	.	14	jm1=j-1;
0.000	30000	.	15	jp1=j+1;
0.029	30000	.	16	for k=2:(sz-1)
0.060	3000000	.	17	km1=k-1;
0.083	3000000	.	18	kp1=k+1;
1.537	3000000	.	19	newplate(j,k)=(plate(jm1,km1)/2+plate(jm1,k)+...
			20	plate(jm1,kp1)/2+plate(j,km1)+plate(j,kp1)+...
			21	plate(jp1,km1)/2+plate(jp1,k)+plate(jp1,kp1)/2)/6;
0.116	3000000	.	22	end
0.001	30000	.	23	end
0.028	300	.	24	plate=newplate;
0.060	300	.	25	if (0==rem(i,5))
0.050	60	x	26	set(im,'cdata',plate)
1.740	60	x	27	drawnow
	60	.	28	end
	300	.	29	end
0.000	1	.	30	elapt=toc;
0.280	1	x	31	close
			32	

图 7.9-8

7.10面向对象编程

7.10.1概念综述

7.10.1.1　类和对象

7.10.1.2　面向对象编程的内涵

（1）创建类目录
（2）选定待建类的数据结构
（3）对象构造函数
（4）显示函数
（5）与其他类之间的转换函数
（6）其他重载函数和重载运算

7.10.2面向对象编程应用示例

【例 7.10.2-1】本例演示：创建“先进先出”FIFO 队列 queue 类的全过程。在本例中，读者应充分注意：构架域（Fields of a structure array）和定义在其上的方法函数（Method function）之间的关系。

（1）

（2）

（3）

[@queue\queue.m]

```
function q=queue(v)
%@QUEUE\QUEUE
% 调用格式
%
%
superiorto('double','sparse','struct','cell','char','inline','sym');
                                          %   <6>
if nargin>1;error('Too many arguments.');end;
if nargin==0                              %
   q.value=[];
   q.name='';
   q=class(q,'queue');
elseif isa(v,'queue');                    %
   q=v;                                       %
else                                  %
   q.value=v;                                 %
   q.name=inputname(1);
   if isempty(q.name)
      q.name=['(' class(v) ')'];
   end
   q=class(q,'queue');                        %   <20>
end
```

（4）

[@queue\display.m]

```
function display(q,ki,kj)
%QUEUE\DISPLAY
% 调用格式
%
%
%
if nargin==0;error('缺少输入宗量，即被显示对象 ！');end
switch nargin
case 1
```

```
    [m,n]=size(q);
    vname=inputname(1);
    if isempty(vname)
       fprintf('ans=\n');
    elseif fprintf('%s=\n',vname);
    end;
    if isempty(q)
       fprintf('    [ empty  ')                                         %<17>
       fprintf('%s',class(q))                                           %<18>
       fprintf(' ]\n\n');                                               %<19>
    elseif m*n==1;
       fprintf('    %s: ',q.name);
       disp(q.value);
       fprintf('\n');
    else
       fprintf('    [ %d*%d ',m,n]                                      %<25>
       fprintf('%s',class(q))                                           %<26>
       fprintf(' ]\n\n');                                               %<27>
    end
case 2
   disp(['The content of ',inputname(1),'(',int2str(ki),')'])
   disp(['is a ''',class(q(ki).value),''' object'])
   fprintf('  %s=\n',q(ki).name);
   disp(q(ki).value);
   fprintf('\n');
case 3
   disp(['The content of ',inputname(1),'(',int2str(ki),',',int2str(kj),')'])
   disp(['is a ''',class(q(ki,kj).value),''' object'])
   fprintf('   %s=\n',q(ki,kj).name);
   disp(q(ki,kj).value);
   fprintf('\n');
end
```

（5）

[@queue\isempty.m]

```
function f=isempty(q)
%@QUEUE\ISEMPTY
f=0;
[m,n]=size(q);
if m*n==1;
   if isempty(q.value) & isempty(q.name)                              %<6>
      f=1;
   end;
end;
```

（6）

[@queue\comein.m]

```
function q=comein(p,varargin)
% @QUEUE\COMEIN
% 调用格式
%
```

```
%
%
if nargin<2 error('comein needs at least two arguments.');end;
if ~isa(p,'queue') error([inputname(1),' is not a queue']);end;
q0=p;
qzzy=class(p);                  %  <10>
for i=1:length(varargin)
   temp=varargin{i};
   s=eval([qzzy,'(temp)']);     %  <13>
   s.name=inputname(i+1);
   if isempty(s.name)
      s.name=['(' class(temp) ')'];
   end
   if isempty(q0)
      q0=s;
   else
      q0=[q0 s];
   end
end
if nargout==0;
   assignin('caller',inputname(1),q0);
   evalin('caller',inputname(1));
else
   q=q0;
end
```

[@queue\goout.m]

```
function [n,v,q]=goout(p)
% @QUEUE\GOOUT
% 调用格式
%
%
%
%
%
if nargin==0 ;error('No queue specifide.');end;
if nargout>3;error('Too many output arguments.');end;
if nargin>1 error('Too many input arguments.');end;
if ~isa(p,'queue');error([inputname(1),' is not a queue.']);end;
if isempty(p)
   q1=p;
else
   [m,n]=size(p);
   v1=p(1).value;n1=p(1).name;
   if m*n==1
      q1=queue;
   else
      q1=p(2:end);
   end
end
if nargout<3;
```

```
   assignin('caller',inputname(1),q1);
end;
if nargout==0,
   evalin('caller',inputname(1));
end
if nargout>=1;v=v1;end;
if nargout>=2;n=n1;end;
if nargout==3;q=q1;end;
```

【例 7.10.2-2】本例的目的：一，检验例 7.10.2-1 所编写的程序的正确性；二，演示所设计的新类是如何被运作的。

（1）

```
qe='Hello!    你好 !';
Q=queue(qe)
Q=
   qe: Hello!    你好 !
```

（2）

```
class(Q)
isobject(Q)
isa(Q,'queue')
ans =
queue
ans =
     1
ans =
     1
```

（3）

```
isempty(Q)
ans =
     0
```

【例 7.10.2-3】本例目的：一，演示"入队"、"离队"函数的调用方法；二，演示@queue\display 显示队列具体元素细节的功能。

（1）

```
a=[1,2,3;4,5,6];b{1}='This';b{2}=' is ';b{3}='a cell array';
comein(Q,a,b)
Q=
   [ 1*3 queue ]
```

（2）

```
display(Q,2)
The content of Q(2)
is a 'double' object
  a=
     1      2      3
     4      5      6
```

（3）

```
[nn,vv,QQ]=goout(Q)
nn =
qe
vv =
Hello!    你好 !
QQ=
```

```
   [ 1*2 queue ]
```

（4）

```
display(QQ,1,2)
The content of QQ(1,2)
is a 'cell' object
   b=
    'This'     ' is '     'a cell array'
```

【例 7.10.2-4】利用指令 methods 可以获知对任何类定义的（在类目录上的）所有方法函数。

```
methods queue

Methods for class queue:

comein    display  goout    isempty  queue
```

7.10.3 重载运算

7.10.4 继承性及其应用

7.10.4.1 继承概念

7.10.4.2 class 函数调用格式汇总

7.10.4.3 利用继承性创建子类的示例

【例 7.10.4.3-1】把例 7.10.2-1 构成的队列作为父类，利用继承性，创建 stack 堆栈子类。

（1）

```
mkdir('d:\matlab6p5\work','@stack')

cd d:\matlab6p5\work\@stack
```

（2）

[@stack\stack.m]

```
function ST=stack(v)
% 调用格式
%
%
if nargin>1;error('Too many arguments.');end;
if nargin==0
   Q=queue;
   s.value=[];
   s.name='';
elseif isa(v,'stack');
   s=v;
   Q=queue(evalin('caller',inputname(1)));
else
   s.value=v;
   s.name=inputname(1);
   if isempty(s.name)
```

```
      s.name=['(' class(v) ')'];
   end
   Q=queue(evalin('caller',inputname(1)));
end
ST=class(s,'stack',Q);
```

【例 7.10.4.3-2】本例目的之一是：检查上例构造函数设计的正确性。目的之二是：观察堆栈关于队列的显示，类别判断和为 “ 空 ” 判断性质的继承。

（1）

```
AA=' 继承性 ';
ST=stack(AA)
ST=
   (char):  继承性
```

（2）

```
class(ST)
ans =
stack
```

（3）

```
isa(ST,'stack')
isa(ST,'queue')
ans =
     1
ans =
     1
```

（4）

```
isempty(ST)
ans =
     0
```

【例 7.10.4.3-3】本例通过堆栈类对象的 “ 压入 ” 和 “ 弹出 ” 操作，进一步观察继承性。

（1）

```
BB=1:6;CC=sym('x^2+4*x');
comein(ST,BB,CC)
ST=
   [ 1*3 stack ]
```

（2）显示堆栈中第三元素的内容

```
display(ST,3)
The content of ST(3)
is a 'sym' object
  CC=
x^2+4*x
```

（3）从堆栈弹出元素

```
[Name1,Value1,ST_1]=goout(ST)
Name1 =
(char)
Value1 =
 继承性
ST_1=
   [ 1*2 stack ]
```

第八章　SIMULINK 交互式仿真集成环境

8.1 引导

SIMULINK 是一个进行动态系统建模、仿真和综合分析的集成软件包。它可以处理的系统包括：线性、非线性系统；离散、连续及混合系统；单任务、多任务离散事件系统。

在 SIMULINK 提供的图形用户界面 GUI 上，只要进行鼠标的简单拖拉操作就可构造出复杂的仿真模型。它外表以方块图形式呈现，且采用分层结构。从建模角度讲，这既适于自上而下（Top-down）的设计流程（概念、功能、系统、子系统、直至器件），又适于自下而上（Bottum-up）逆程设计。从分析研究角度讲，这种 SIMULINK 模型不仅能让用户知道具体环节的动态细节，而且能让用户清晰地了解各器件、各子系统、各系统间的信息交换，掌握各部分之间的交互影响。

在 SIMULINK 环境中，用户将摆脱理论演绎时需做理想化假设的无奈，观察到现实世界中摩擦、风阻、齿隙、饱和、死区等非线性因素和各种随机因素对系统行为的影响。在 SIMULINK 环境中,用户可以在仿真进程中改变感兴趣的参数,实时地观察系统行为的变化。由于 SIMULINK 环境使用户摆脱了深奥数学推演的压力和烦琐编程的困扰，因此用户在此环境中会产生浓厚的探索兴趣，引发活跃的思维，感悟出新的真谛。

在 MATLAB6.x 版中，可直接在 SIMULINK 环境中运作的工具包很多，已覆盖通信、控制、信号处理、DSP、电力系统等诸多领域，所涉内容专业性极强。本书无意论述涉及工具包的专业内容，而只是集中阐述：SIMULINK 的基本使用技法和相关的数值考虑。

节 8.1 虽是专为 SIMULINK 初学者写的，但即便是熟悉 SIMULINK 以前版本的读者也值得快速浏览这部分内容，因为新版的界面、菜单、工具条、模块库都有较大的变化。第 8.2 节比较详细地阐述建模的基本操作：通用模块的具体化设置、信号线勾画、标识、模型窗参数设置。这部分内容是进一步深入的前提。从第 8.3 节起，由浅入深地讲述 SIMULINK 对各种数学、工程问题的建模、仿真和分析的基本方法。

本章采用“算例”作为主体，配以适量的归纳性表述。本章包含了 34 个“尽量简单”又“独立完整”的“典型”算例，而这正是 SIMULINK 在线 PDF 文件之所缺。读者通过“手、眼、脑”并用地练习算例，掌握 SIMULINK 的一般使用规则和操作技法。

鉴于 SIMULINK 的本质，本节算例必定涉及数学、物理、和若干工程考虑。本书已采取“无量纲记述”、“注释”等措施使算例尽可能易读易懂，读者只要稍微耐心，就可以从这些有背景的内容体验到 SIMULINK 仿真之细腻和切实，从这些带背景性的算例品出 SIMULINK 的精妙之处。

本章内容已在 MATLAB6.5 基础上进行全面更新，变动最大的是第 8.4.3 节。此外，为适应读者应用水平的提高，新增了第 8.8 节，论述 S 函数模块的创建和使用。

8.1.1 SIMULINK 的安装

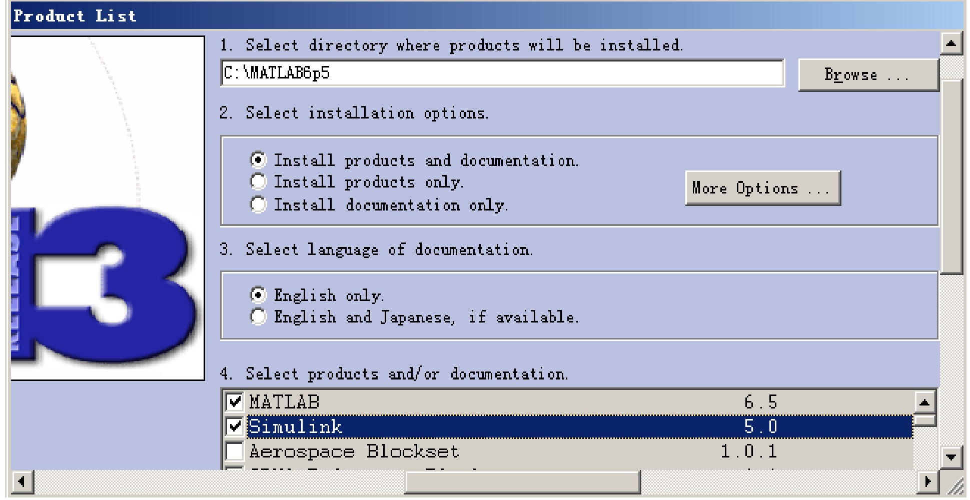


图 8.1.1-1

8.1.2 SIMULINK 入门

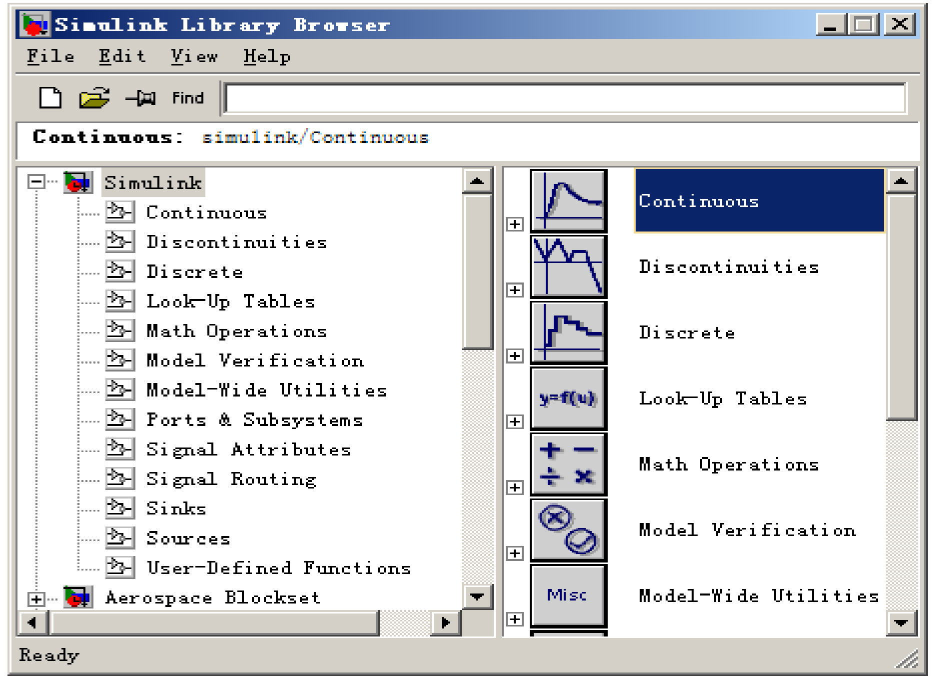


图 8.1.2-1

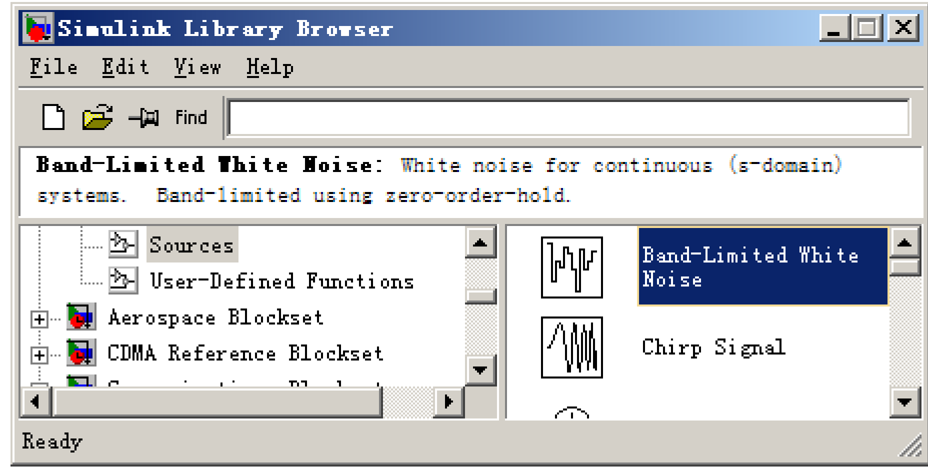


图 8.1.2-2

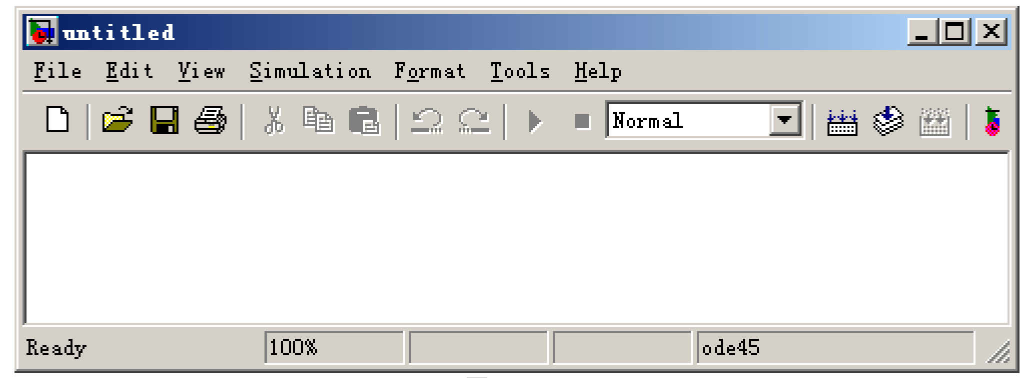


图 8.1.2-3

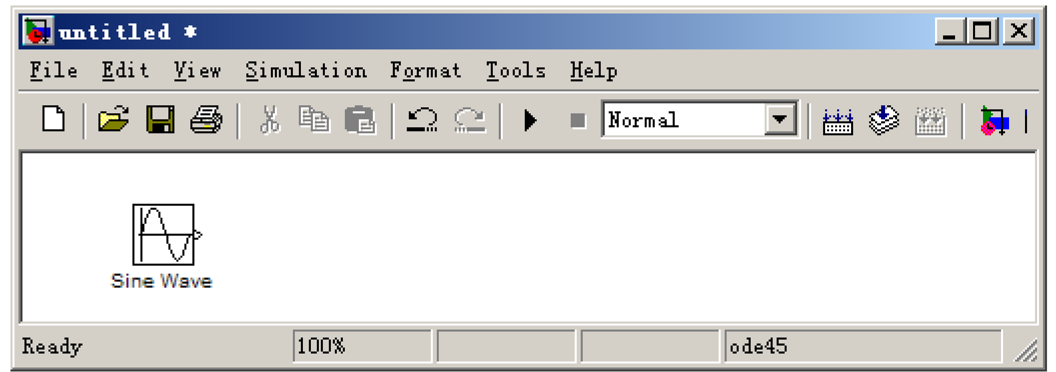


图 8.1.2-4 模型创建中的模型窗一

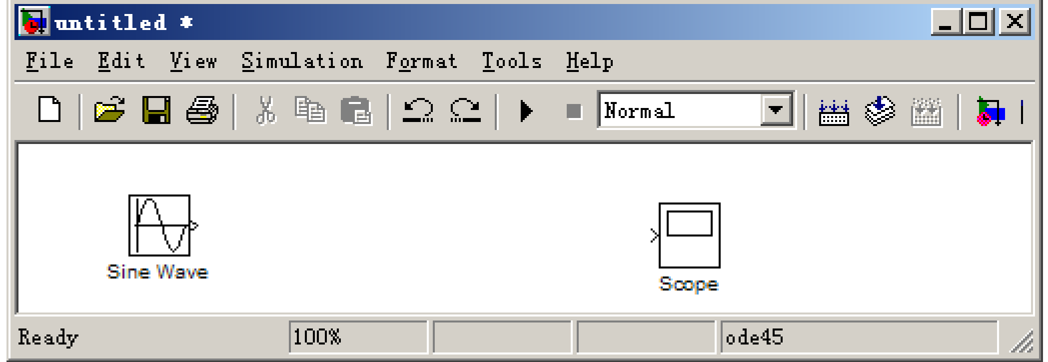


图 8.1.2-5

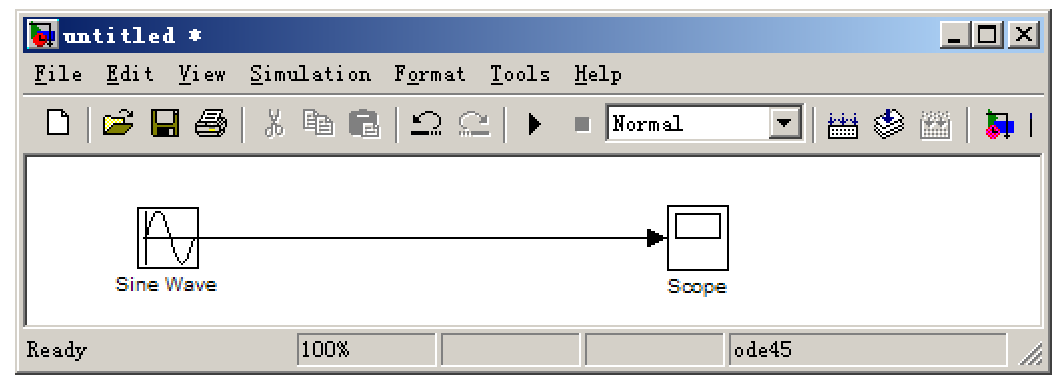


图 8.1.2-6

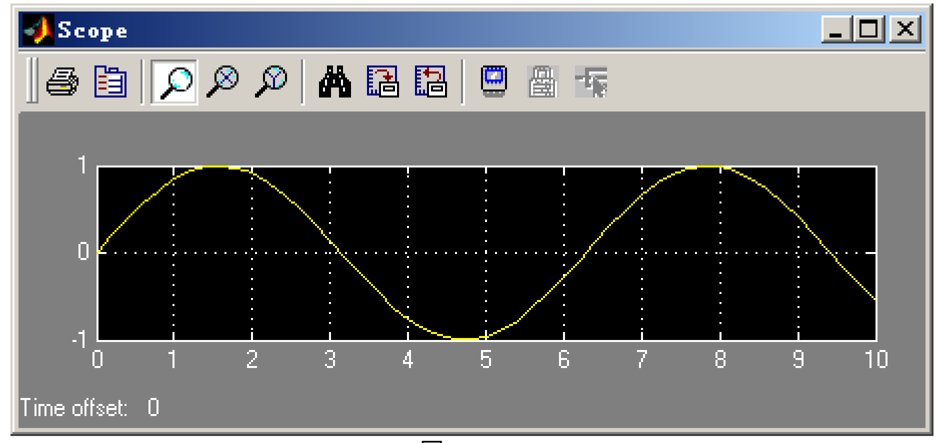


图 8.1.2-7

8.1.3 SIMULINK 库浏览器界面

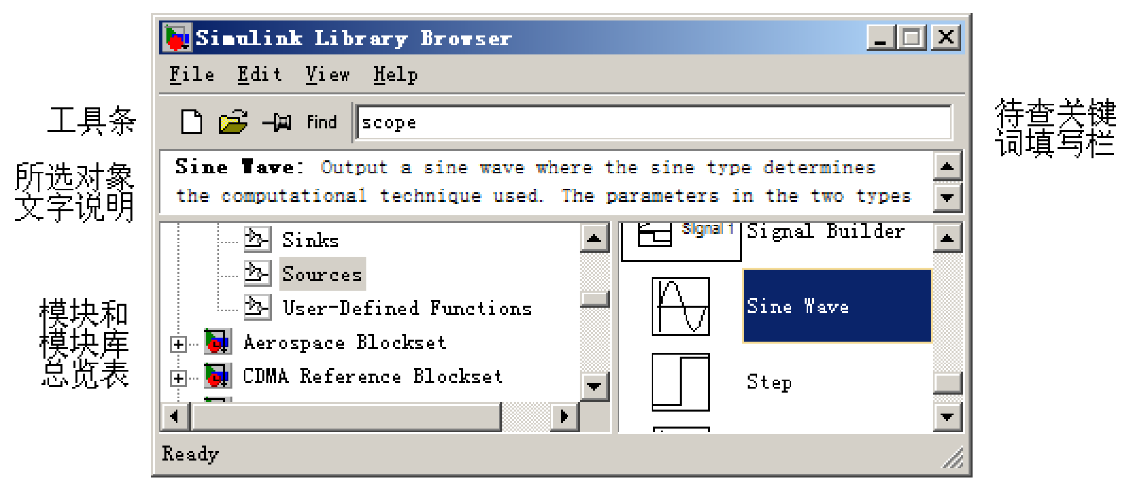


图 8.1.3-1

8.1.4 SIMULINK 模型窗的组成

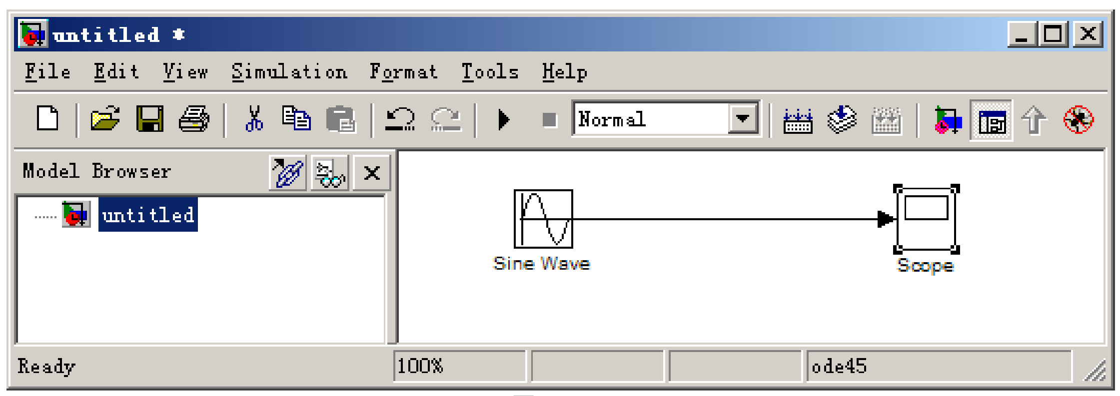


图 8.1.4-1

8.2 模型的创建

8.2.1 模型概念和文件操作

8.2.1.1 SIMULINK 模型是什么

8.2.1.2 模型文件的操作

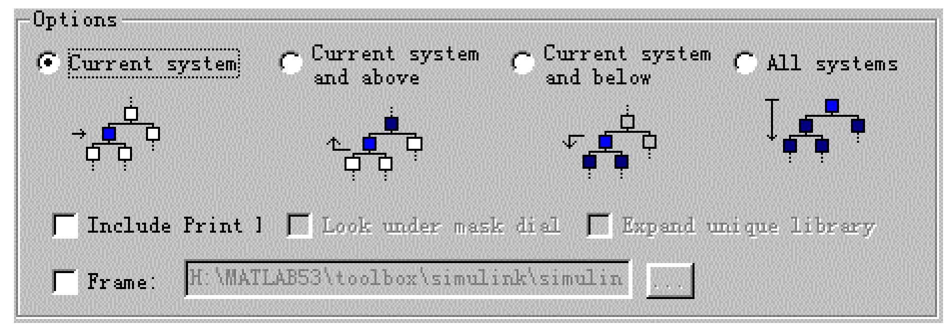


图 8.2.1.2-1

8.2.2 模块操作

8.2.2.1 模块的基本操作

（1）模块的选定

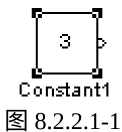


图 8.2.2.1-1

- 选定单个模块的操作方法：
- 选定多个模块的操作方法：

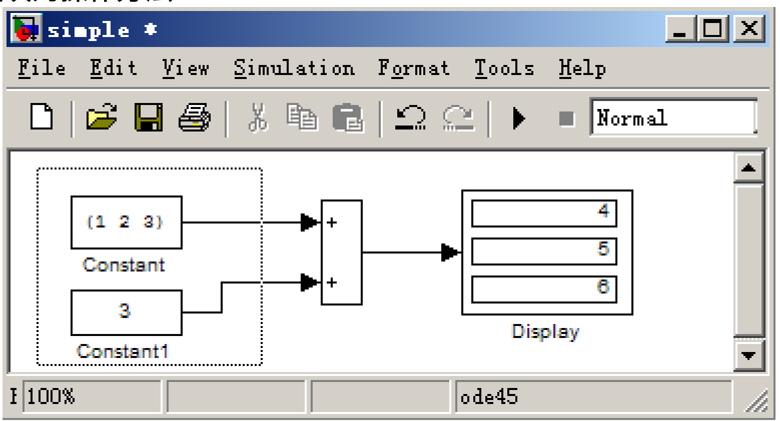


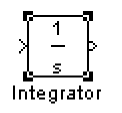
图 8.2.2.1-2

（2）模块的复制

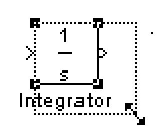
（3）模块的移动

（4）模块的删除

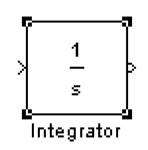
（5）改变模块大小



(a) 原尺寸



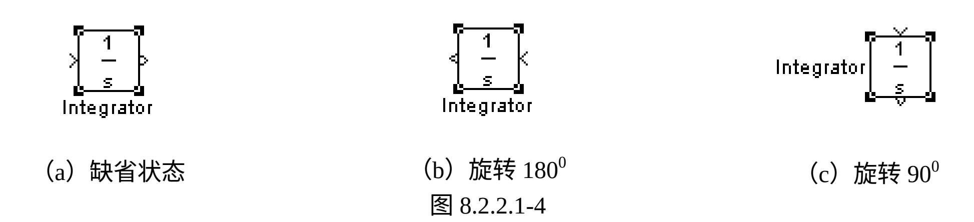
(b) 拖动边框



(c) 新尺寸

图 8.2.2.1-3

（6）模块的旋转



（a）缺省状态　　　（b）旋转 180^0　　　（c）旋转 90^0

图 8.2.2.1-4

（7）模块名的操作

（8）模块的阴影效果

8.2.2.2 向量化模块和标量扩展

（1）向量化模块

（2）标量扩展

【例 8.2.2.2-1】演示“示波”模块的向量显示能力。

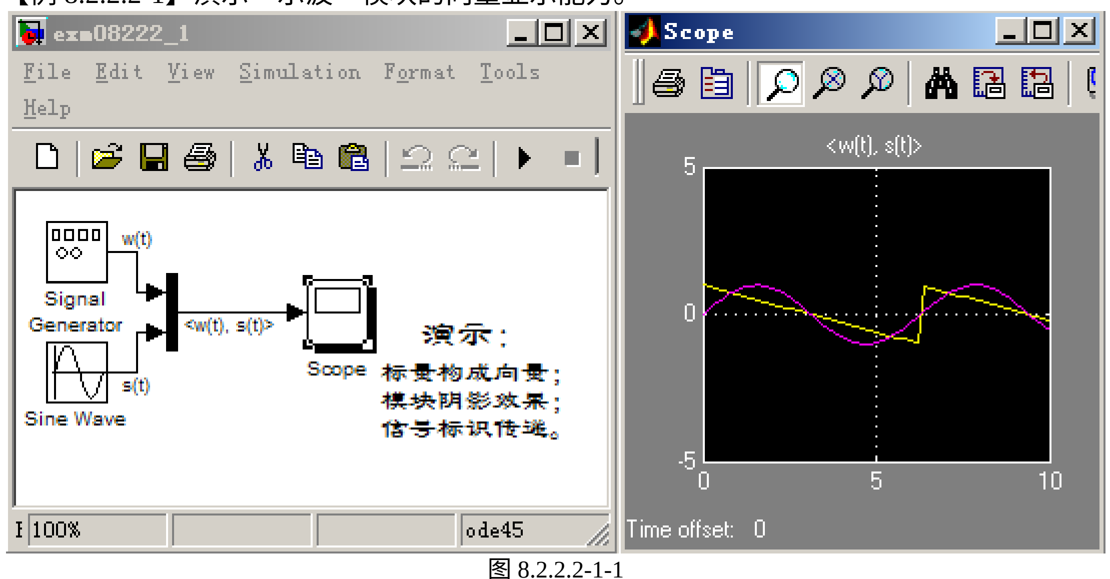


图 8.2.2.2-1-1

【例 8.2.2.2-2】演示“求和”模块的向量处理能力：输入扩展。

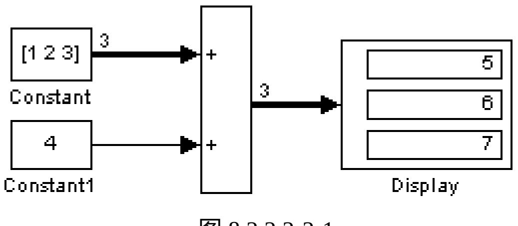


图 8.2.2.2-2-1

【例 8.2.2.2-3】演示“增益”模块的向量处理能力：参数扩展。

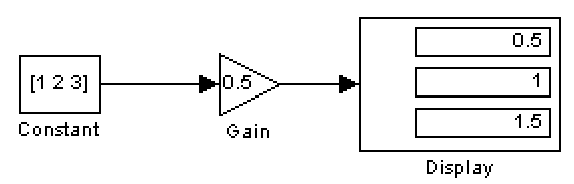


图 8.2.2.2-3-1

8.2.2.3 参数设置

8.2.3 信号线操作

8.2.3.1 产生连线

(1) 水平或垂直连线的产生
(2) 斜连线的产生
(3) 连线的移动和删除

8.2.3.2 信号线的分支和折曲

(1) 分支的产生
(2) 信号线的折曲
(3) 折点的移动

8.2.3.3 信号线宽度显示

8.2.3.4 彩色显示信号线

8.2.3.5 插入模块

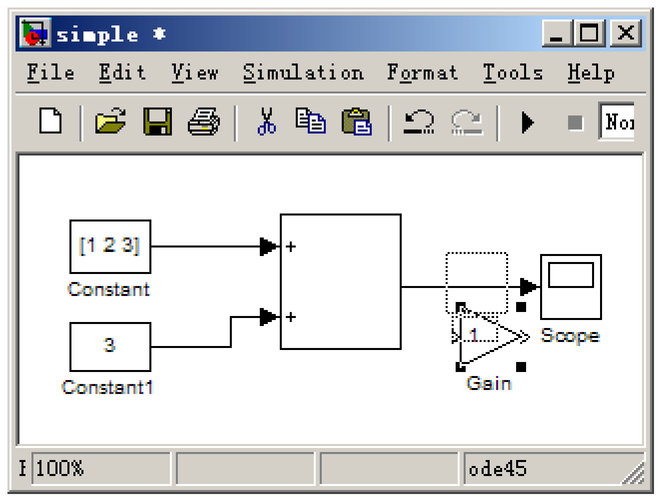


图 8.2.3.5-1

8.2.3.6 信号线标识（label）

【例 8.2.3.6-1】演示：信号线标识的传播

8.2.4 对模型的注释

(1) 模型注释的创建

(2) 注释位置的移动

(3) 注释文字的字体控制

8.2.5 常用的 Sourse 库信源

【例 8.2.5-1】如何调用 MATLAB 工作空间中的信号矩阵作为模型输入。本例所需的输入为

$$u(t)=\begin{cases} t^2 & 0\le t<T \\ (2T-t)^2 & T\le t<2T \\ 0 & else \end{cases}$$ 。

（1）

[sourec 0825_1.m]

```
function TU=source0825_1(T0,N0,K)
t=linspace(0,K*T0,K*N0+1);
N=length(t);
u1=t(1:(N0+1)).^2;
u2=(t((N0+2):(2*N0+1))-2*T0).^2;
u3(1:(N-(2*N0+2)+1))=0;
u=[u1,u2,u3];
TU=[t',u'];
```

（2）

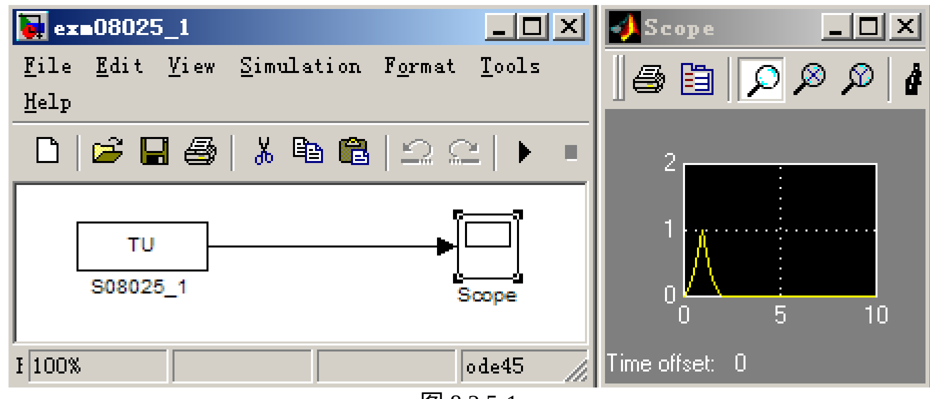


图 8.2.5-1

（3）

（4）

```
TU=source0825_1(1,100,4);
```

（5）

8.2.6 常用的 Sink 库信宿

8.2.6.1 库信宿一览表

8.2.6.2 示波器

（1）示波器的用途
（2）示波器窗的工具条
（3）示波器纵坐标范围的手工设置
（4）示波器横坐标的设置
（5）把示波器数据送入 MATLAB 工作空间
（6）多信号显示区设置
（7）设置为游离示波器

8.2.7 仿真的配置

8.2.7.1 解算器参数的设置（Solver）

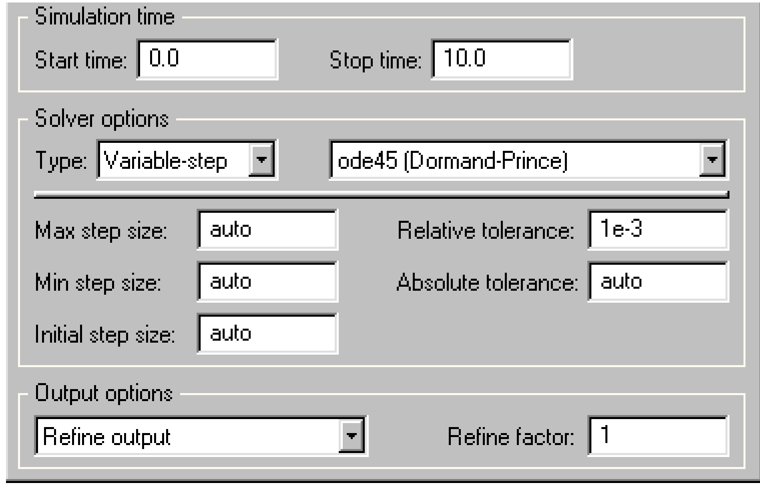


图 8.2.7.1-1

8.2.7.2 仿真数据的输入输出设置（Workspace I/O）

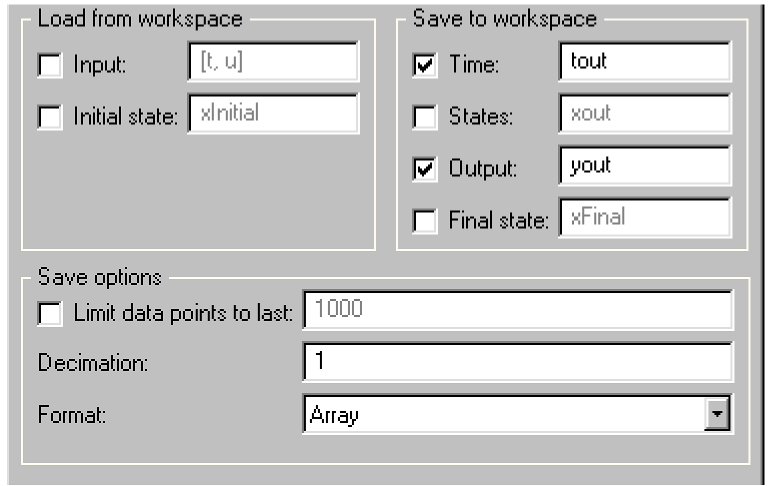


图 8.2.7.2-1

8.2.7.3 仿真中异常情况的诊断（Diagnostics）

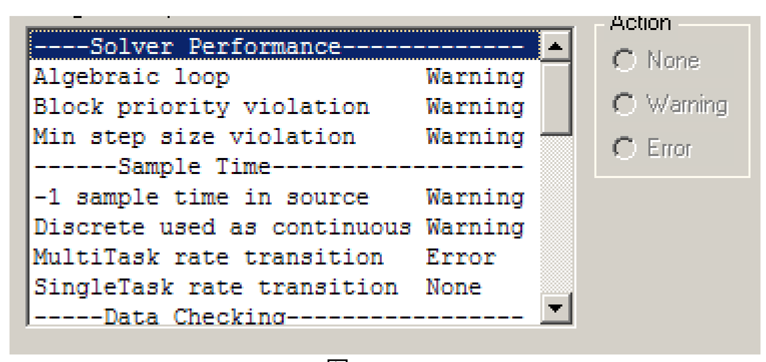


图 8.2.7.3-1

8.3 连续系统建模

8.3.1 线性系统

8.3.1.1 积分模块的功用

【例 8.3.1.1-1】复位积分器的功用示例。

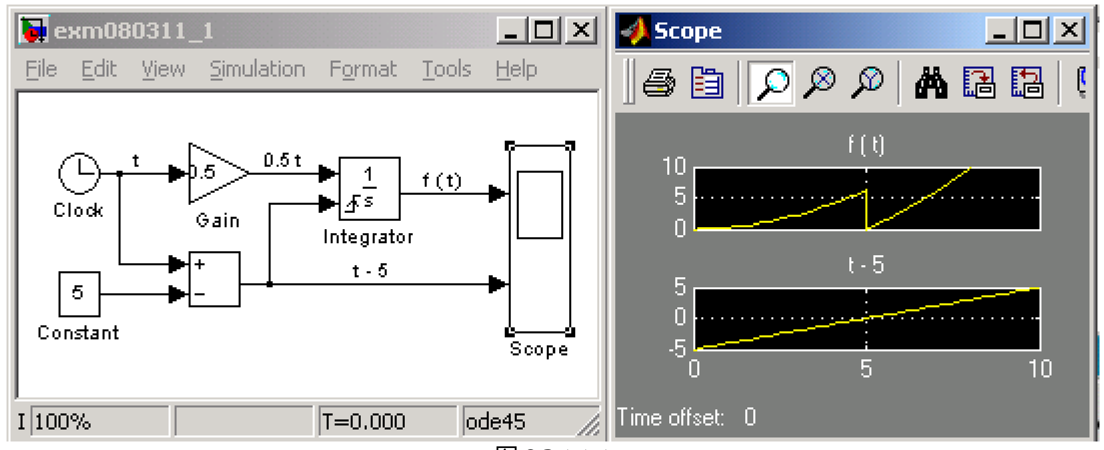


图 8.3.1.1-1

8.3.1.2 积分模块直接构造微分方程求解模型

【例 8.3.1.2-1】假设从实际自然界（力学、电学、生态等）或社会中，抽象出有初始状态为 0 的二阶微分方程 $x'' + 0.2x' + 0.4x = 0.2u(t)$，$u(t)$ 是单位阶跃函数。本例演示如何用积分器直接构搭求解该微分方程的模型。

（1）

（2）

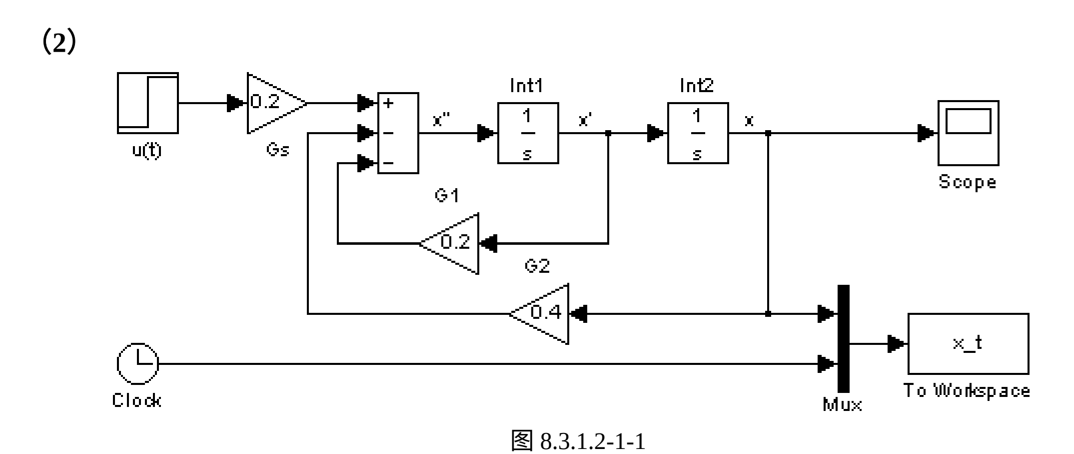


图 8.3.1.2-1-1

（3）仿真操作

（4）保存在 MATLAB 工作空间中的数据

```
clf
tt=ScopeData.time;
xx=ScopeData.signals.values;
[xm,km]=max(xx);
plot(tt,xx,'r','LineWidth',4),hold on
plot(tt(km),xm,'b.','MarkerSize',36),hold off
strmax=char('最大值',['t = ',num2str(tt(km))],['x = ',num2str(xm)]);
text(6.5,xm,strmax),xlabel('t'),ylabel('x')
```

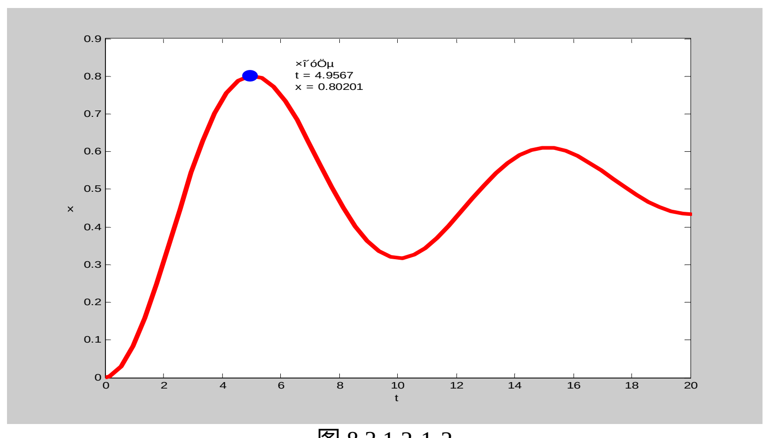


图 8.3.1.2-1-2

8.3.1.3 传递函数模块

【例 8.3.1.3-1】直接利用传递函数模块求解方程(8.3.1.3-1)。

(1)

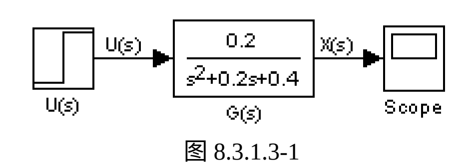


图 8.3.1.3-1

(2)

8.3.1.4 状态方程模块和单位脉冲输入的生成

【例 8.3.1.4-1】假设式(8.3.1.4-1)中的输入函数 u 是单位脉冲函数 $\delta(t)$，研究该系统的位移变化。本例演示：（A）状态方程模块的使用；（B）脉冲函数的生成方法。

(1)

(2)

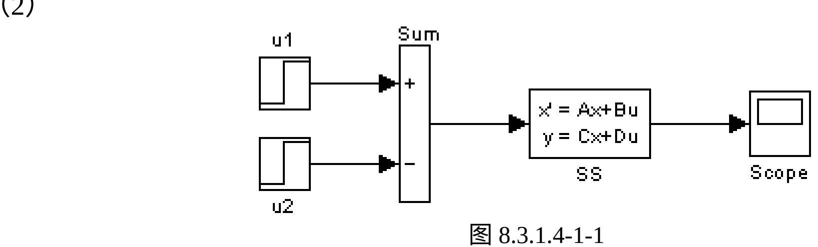


图 8.3.1.4-1-1

(3)

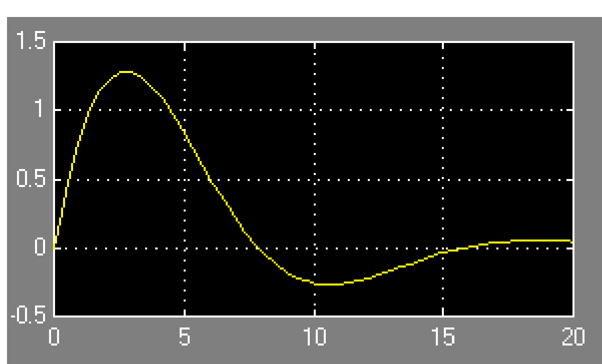


图 8.3.1.4-1-2

8.3.2 非线性系统

8.3.2.1 建立非线性仿真模型的基本考虑

【例 8.3.2.1-1】物理背景：如图 8.3.2.1-1-1 所示喷射动力车的定位控制问题。

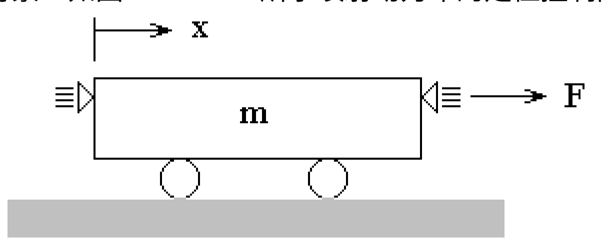


图 8.3.2.1-1-1

（1）

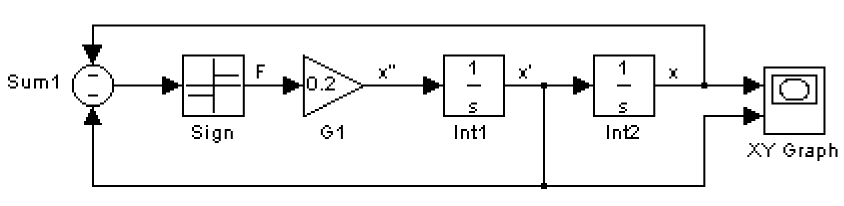


图 8.3.2.1-1-2

（2）

（3）

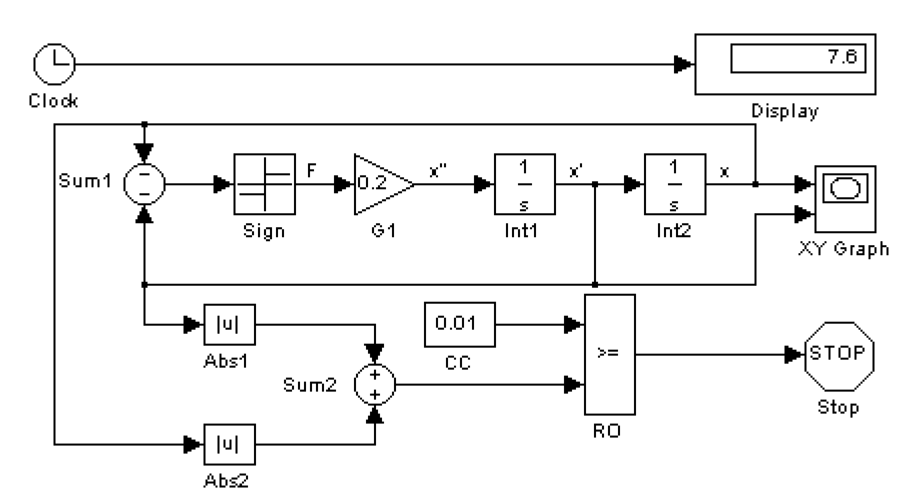


图 8.3.2.1-1-3

（4）

（5）

（6）

```
subplot(1,2,1),plot(xout(:,2),xout(:,1))
grid on,axis([-0.2,1,-1,0.2]),axis square
xlabel('\fontsize{14}位移'),ylabel('\fontsize{14}速度'),
subplot(1,2,2),plot(xout(:,2),xout(:,1))
grid on,axis([-0.1,0.05,-0.05,0.1]),axis square
```

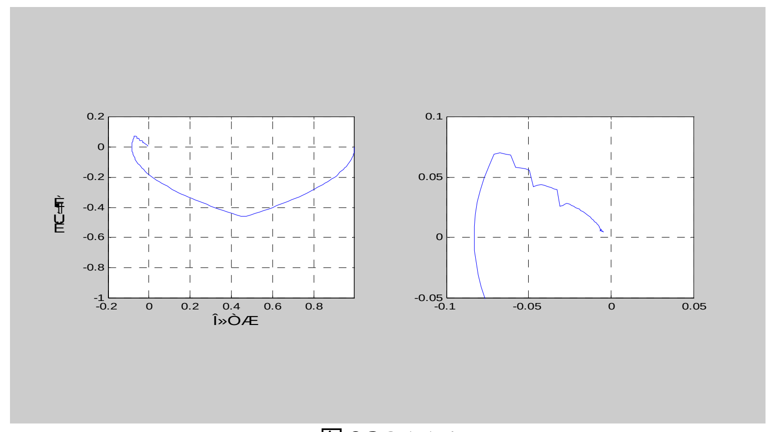


图 8.3.2.1-1-4

8.3.2.2 任意非线性函数模块及其应用

【例 8.3.2.2-1】轿车沿直线山坡路向前行驶。要求设计一个简单的比例放大器，使轿车能以指定的速度运动。本例演示：（A）仿真系统的创建。（B）非线性模块的使用。（C）任意函数模块的应用。(D)体现“ 自下而上 ”的建模方式。（E）本例将作为下面章节多个算例的基础，读者切莫跳略此题。

（1）

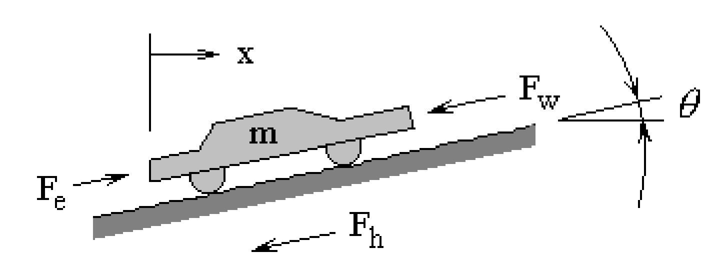


图 8.3.2.2-1-1

（2）

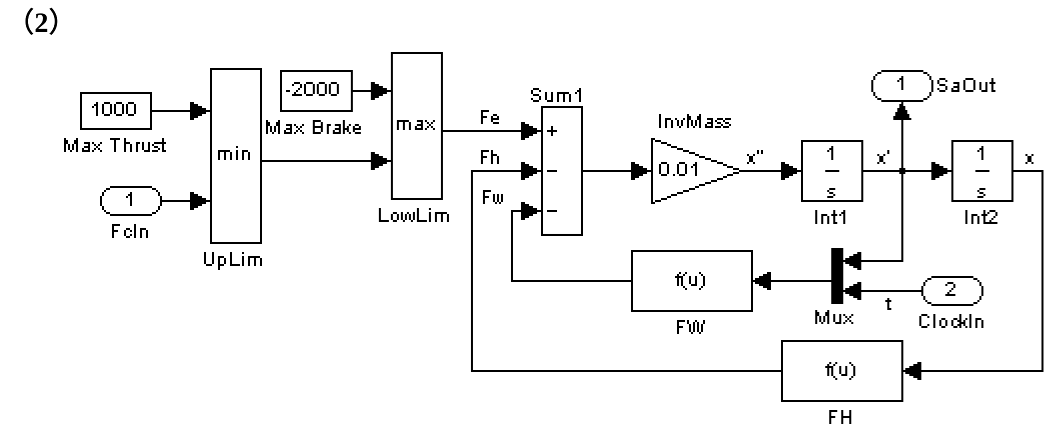


图 8.3.2.2-1-2

（2）

图 8.3.2.2-1-3

(3)

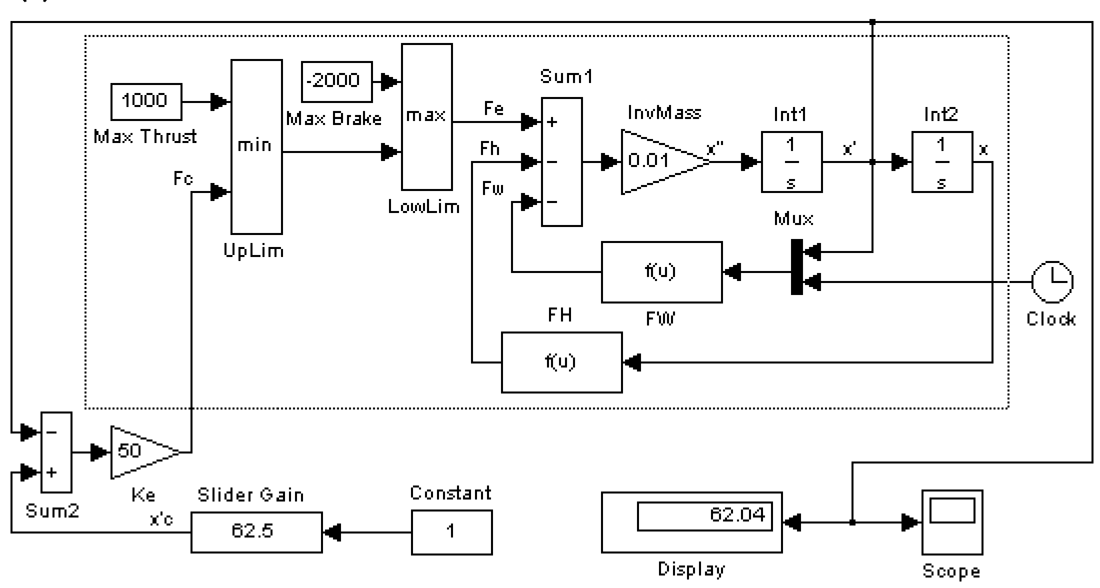


图 8.3.2.2-1-4

(4)

8.4 子系统的创建、装帧及受控执行

8.4.1 简装子系统及其应用

8.4.1.1 创建简装子系统的“ 先有内容后套包装 ”法

【例 8.4.1.1-1】题目的背景和参数与例 8.3.2.2-1 完全相同，要求创建利用比例控制器使轿车的运动速度稳定在期望车速的分层仿真模型。本例演示：如何从非分层模型获得分层模型；创建简装子系统的“ 先有内容后套包装 ”法。

(1)

(2)

(3)

(4)

(5)

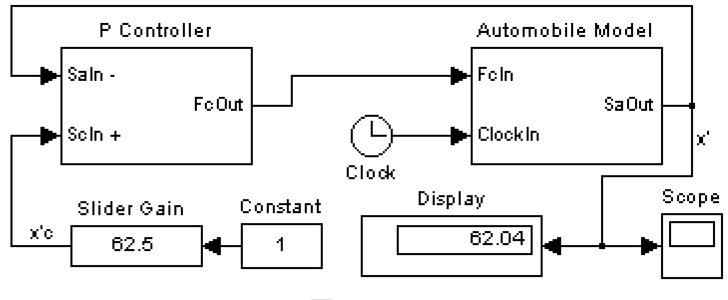


图 8.4.1.1-1

8.4.1.2 创建简装子系统的“ 先有包装后置内容 ”法

【例 8.4.1.2-1】本例演示：如何自上而下构造分层模型；产生简装子系统的“先有包装后置内容”法。
（1）

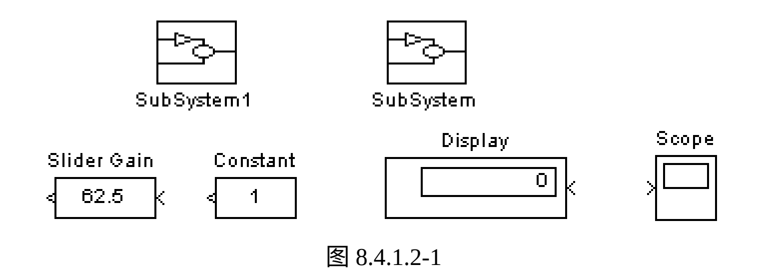


图 8.4.1.2-1

（2）

（3）

（4）

（5）

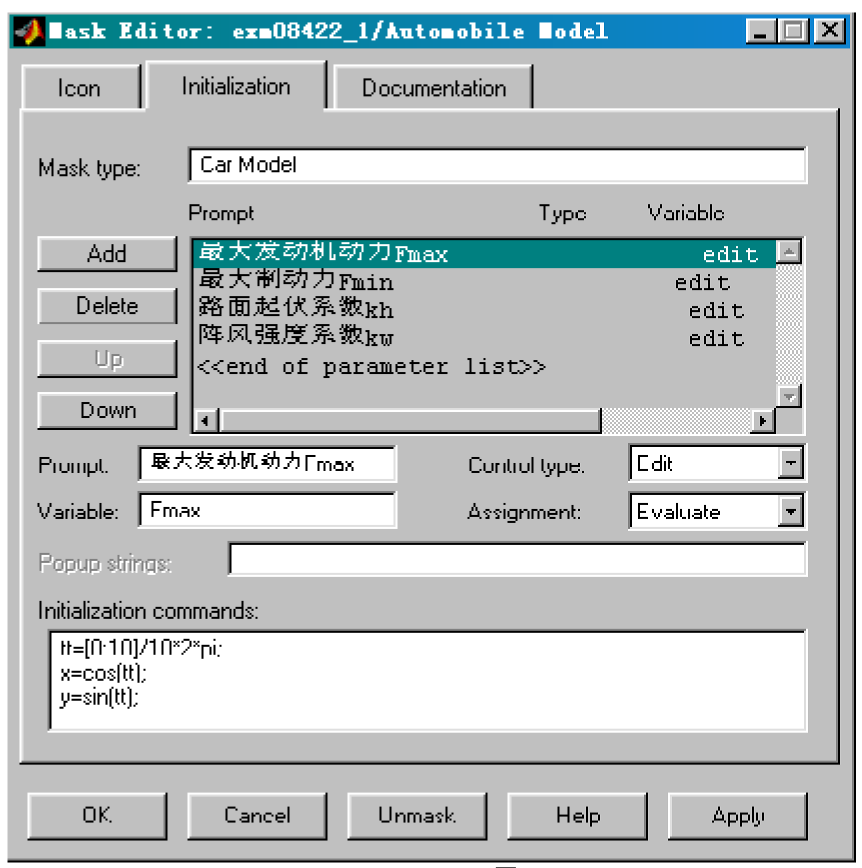
8.4.2 精装子系统

8.4.2.1 精装子系统的制作过程

8.4.2.2 装帧示例

【例 8.4.2.2-1】目标：把图 8.4.1.1-1 所示轿车速度控制模型中的轿车动态模型简装子系统变成精装子系统。
（1）

（2）



块类别区

对话窗参数设置栏排版区

对话窗参数设置栏内容及属性定义区

初始指令区

图 8.4.2.2-1-1

(3)

(4)

(5)

(6)

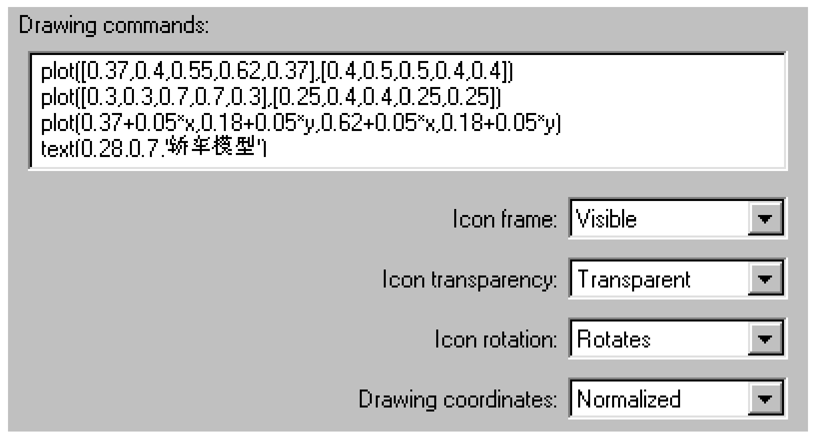


图 8.4.2.2-1-2

(7)

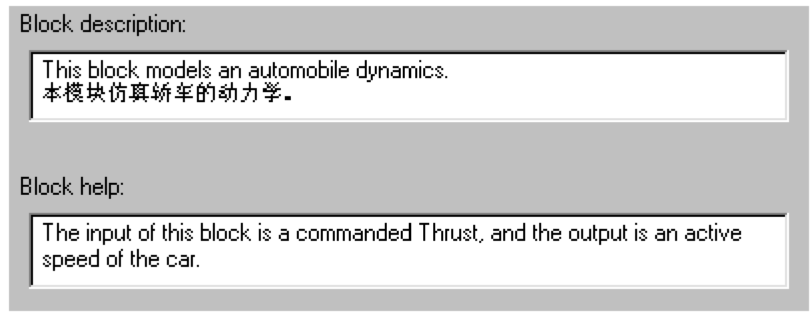


图 8.4.2.2-1-3

（8）

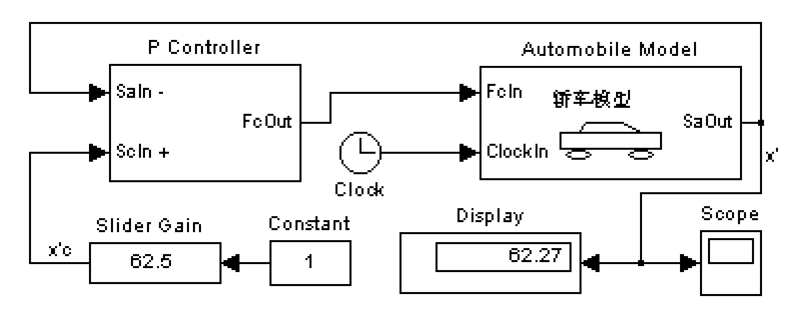


图 8.4.2.2-1-4

8.4.2.3 精装子系统的使用特点

【例 8.4.2.3-1】本例演示：精装子系统参数对话窗的来源和外形特点；如何打开精装子系统自身的“下层”结构模型；精装子系统如何从外界获得参数。

（1）

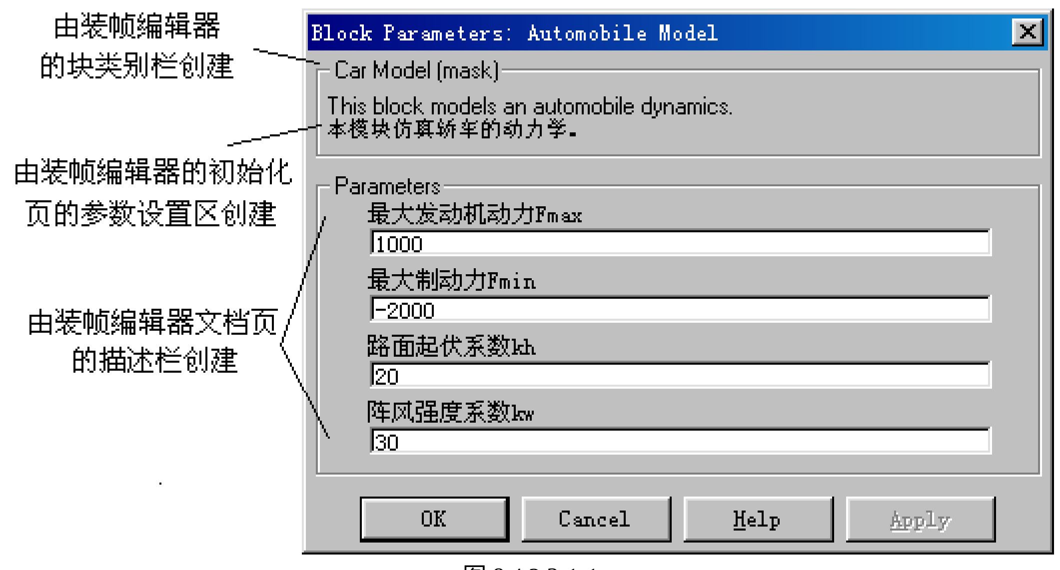


图 8.4.2.3-1-1

（2）

（3）

8.4.3 条件执行子系统

8.4.3.1 使能子系统

【例 8.4.3.1-1】利用使能原理构成一个半波整流器。本例演示使能子系统的创建及工作机理。
(1)
(2)
(3)
(4)

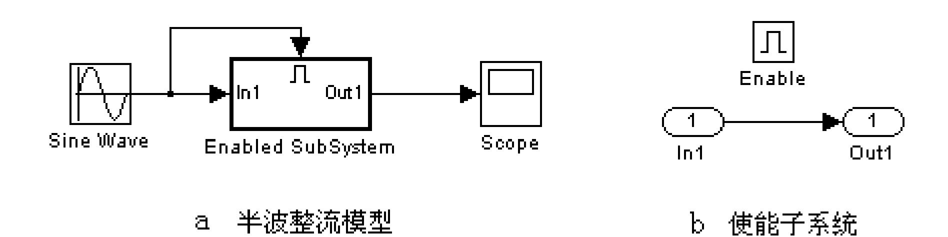


图 8.4.3.1-1-1

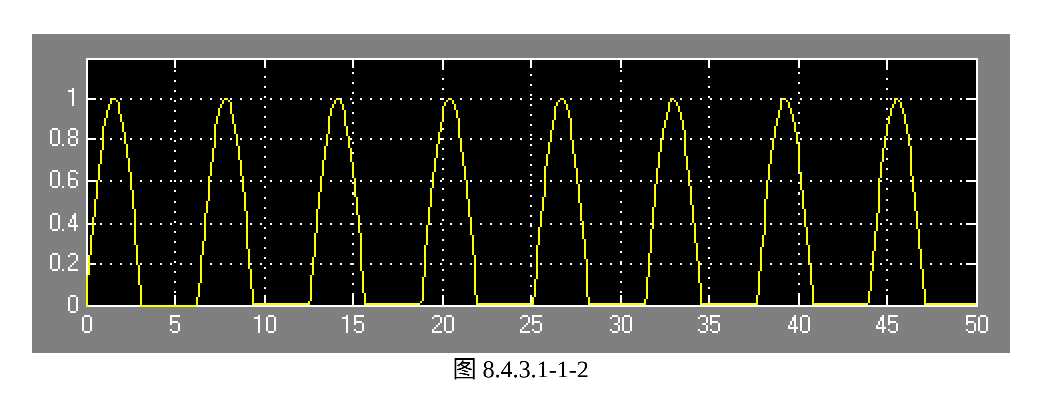


图 8.4.3.1-1-2

【例 8.4.3.1-2】本例演示：在使能子系统中插入滤波模块 $G(s)=\dfrac{0.8}{5s+0.8}$。

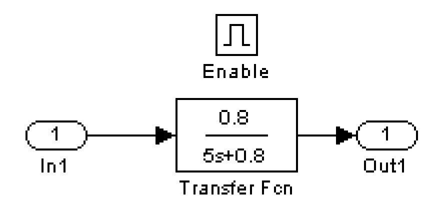


图 8.4.3.1-2-1

(1)
(2)
(3)
(4)

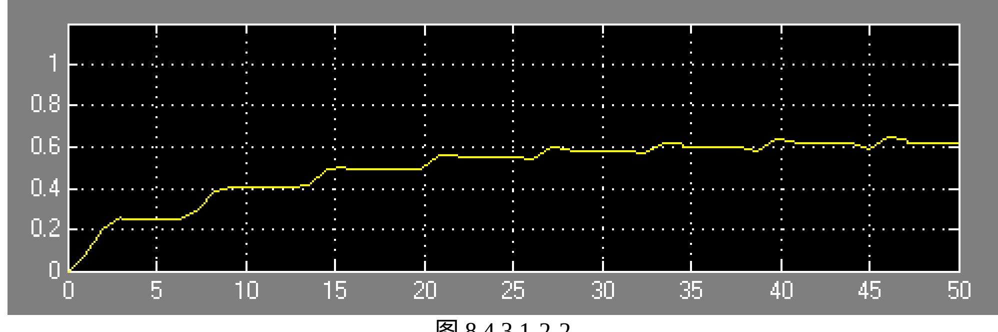


图 8.4.3.1-2-2

8.4.3.2 触发子系统

【例 8.4.3.2-1】利用触发子系统获得零阶保持的采样信号。本例演示：触发子系统工作原理。

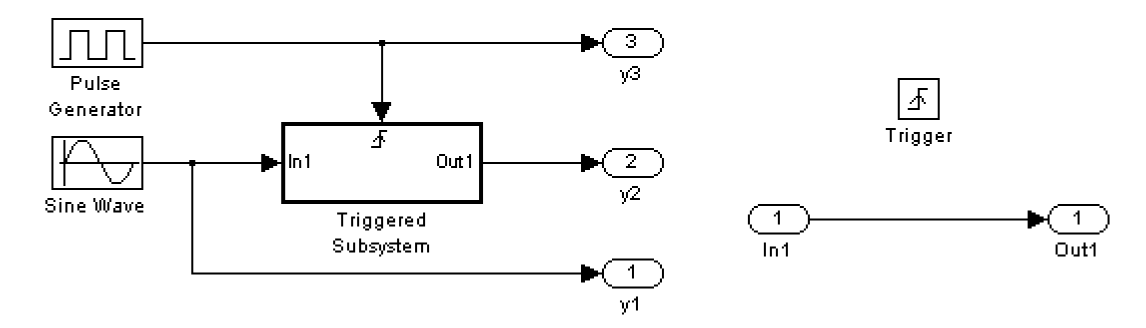


图 8.4.3.2-1-1

（1）

（2）

```
[t,x,y]=sim('exm080432_1',10);
clf,hold on
plot(t,y(:,1),'b')
stairs(t,y(:,2),'r')
stairs(t,y(:,3),'c:'),hold off
axis([0 10 -1.1 1.1]),box on
legend('sinewave','output','trigger',4)
```

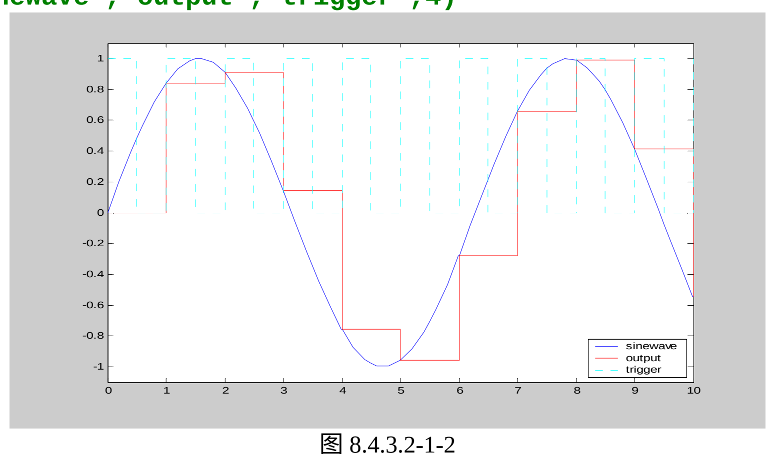


图 8.4.3.2-1-2

8.4.3.3 触发使能子系统

8.4.3.4 使能子系统和出发子系统综合运用示例

【例 8.4.3.4-1】本例是前面例 8.3.2.2-1 , 8.4.1.1-1 , 8.4.2.2-1 的继续，使得汽车速度受两种不同的控制器操纵。具体要求是：（A）当汽车实际速度与期望速度的误差绝对值 $v_{err}=|\dot{x}_c-\dot{x}_d|<2$，且 $\dot{v}_{err}<1$时，将切换为 PI 比例-积分控制器；（B）一旦 PI 控制器被使用，只要仍满足 $v_{err}<5$，那么 PI 将继续起控制作用；（C）除以上情况外，则都使用简单的 P 比例控制器。

（1）

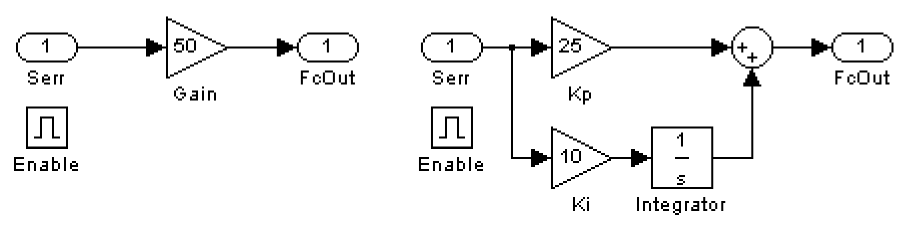


图 8.4.3.4-1-1

(2)

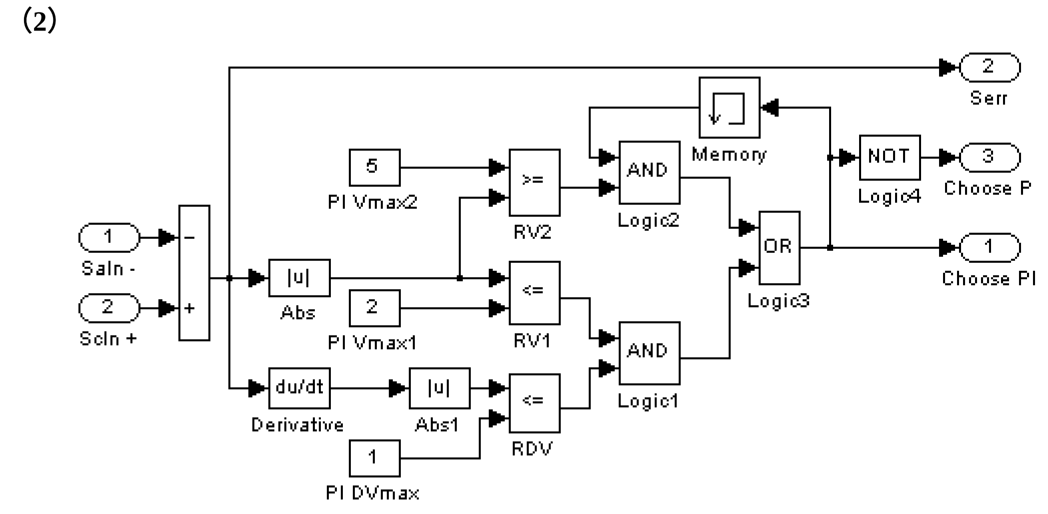


图 8.4.3.4-1-

(3)

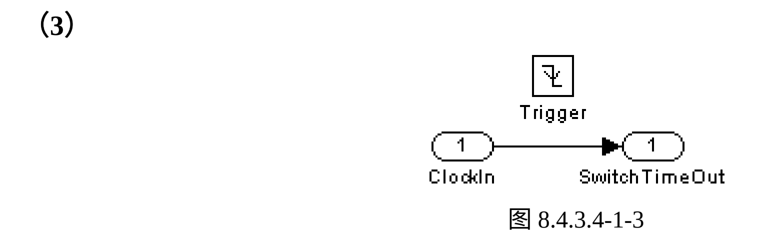


图 8.4.3.4-1-3

(4)

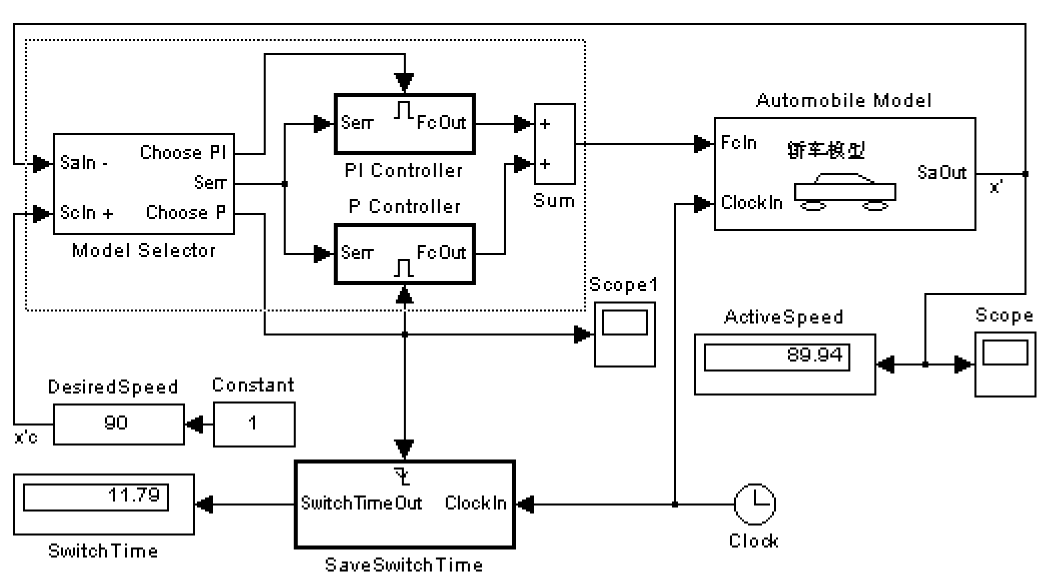


图 8.4.3.4-1-4

（5）

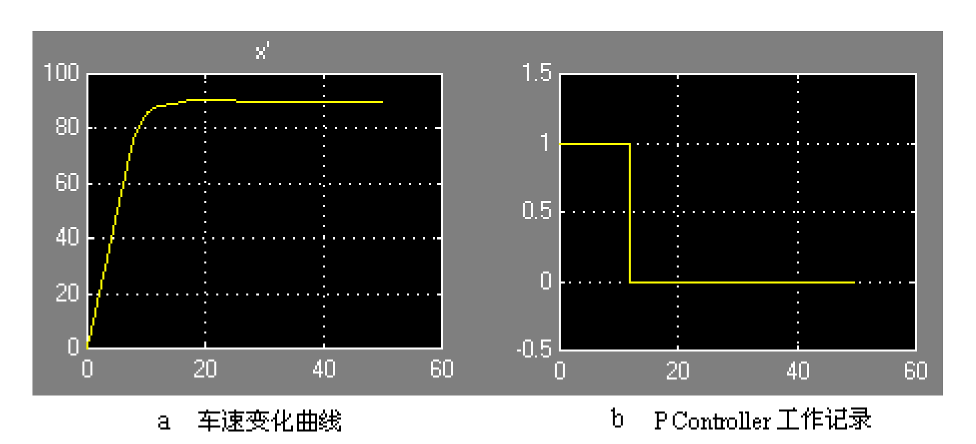


图 8.4.3.4-1-5

8.4.3.5 交替执行子系统

【例 8.4.3.5-1】在例 8.4.3.4-1 中，比例控制器和比例-积分控制器的工作切换是借助 Model Selector 子系统产生的两个输出切换信号 Choose PI 和 Choose P 实现的。本例将演示：如何依靠一个 Choose PI 信号和 merge 汇合模块的配合使用，实现同样的控制器切换。

（1）

（2）

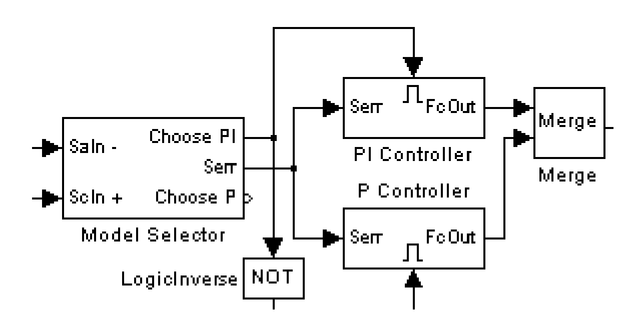


图 8.4.3.5-1

8.5 离散时间系统和混合系统

8.5.1 若干基本模块

（1）单位延迟模块 Unit delay

（2）零阶保持器 Zero-Order hold

（3）传递函数型模块

（4）组合逻辑模块 Combinational logic

（5）离散时间积分器 Discrete-time integrator

【例 8.5.1-1】用组合逻辑模块产生 a,b 的“逻辑和”结果 $c(1)$ 及“逻辑或”结果 $c(2)$。

（1）

表 8.5.1-2

a	b	c(1)	c(2)
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

（2）

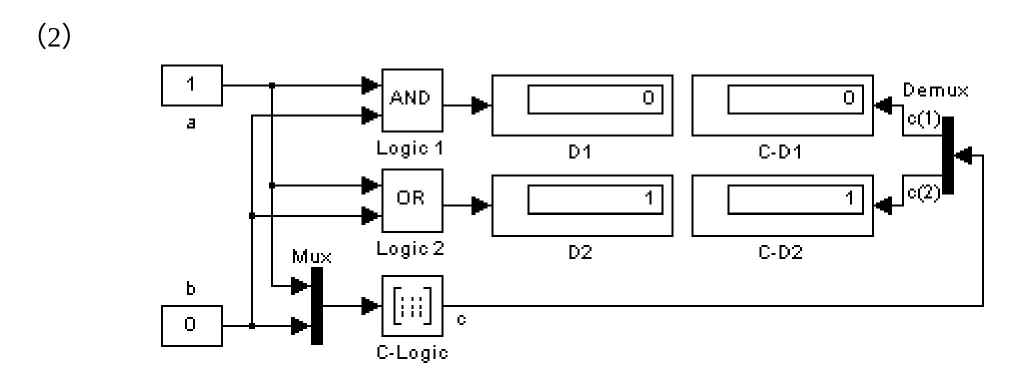


图 8.5.1-1

8.5.2 多速率离散时间系统

【例 8.5.2-1】在离散控制系统中，控制器的更新频率一般低于对象本身的工作频率。而显示系统的更新频率总比显示器的可读速度低得多。假设有某过程的离散状态方程

$$\begin{cases} x_1(k+1) = x_1(k) + 0.1x_2(k) \\ x_2(k+1) = -0.05\sin x_1(k) + 0.094x_2(k) + u(k) \end{cases}$$

式中$u(k)$是输入。该过程的采样周期为0.1秒。控制器应用采样周期为0.25秒的比例控制器；显示系统的更新周期为0.5秒。

（1）

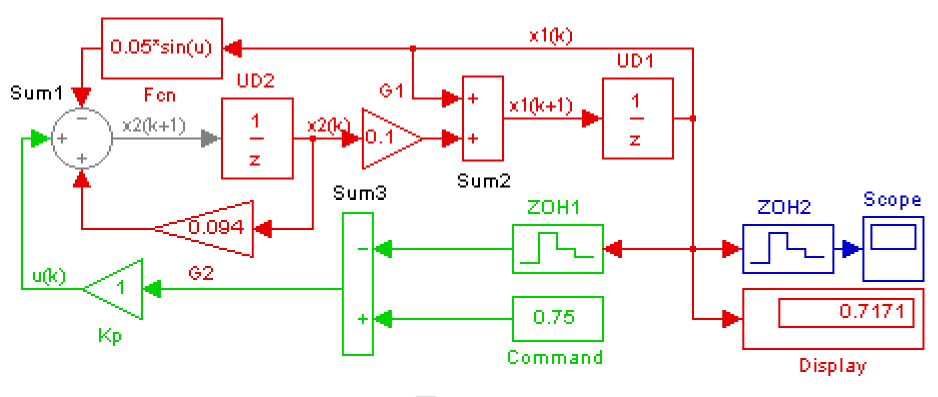


图 8.5.2-1-1

（2）

（3）

```
tt=TX.time;
x1=TX.signals.values;
plot(tt,x1),grid on,
xlabel('kT'),ylabel('x1(kT)')
```

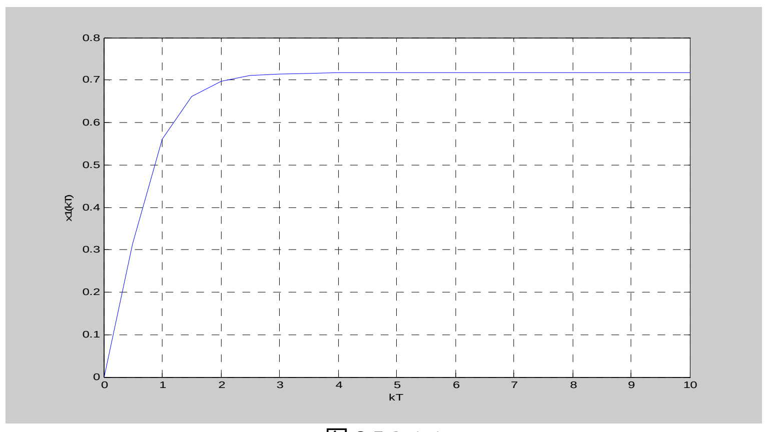


图 8.5.2-1-1

8.5.3 离散-连续混合系统

【例 8.5.3-1】本例是在例 8.4.2.2-1 的基础上进行的。目标是：设计一个离散 PID 控制器子系统对轿车速度进行控制。本例演示：（A）离散 PID 的构成；（B）展示仿真模型在研究控制器各参数影响上的能力。

（1）

（2）

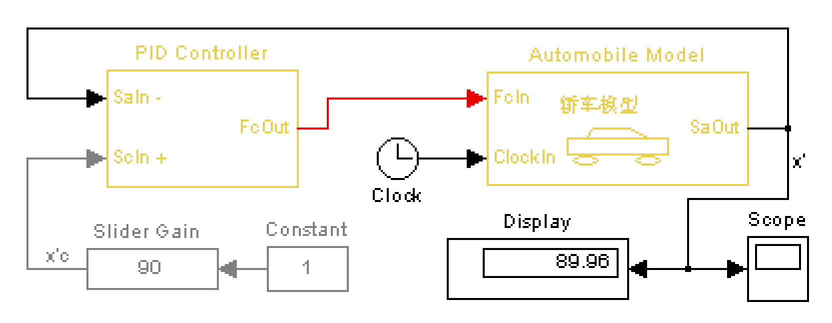


图 8.5.3-1-1

（3）

（4）

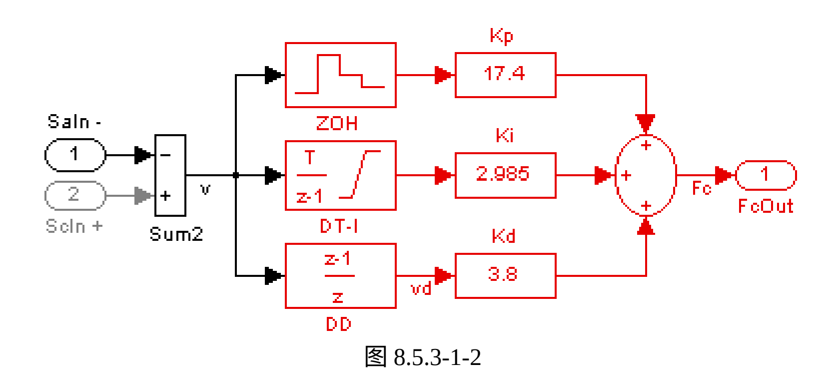


图 8.5.3-1-2

（5）

（6）

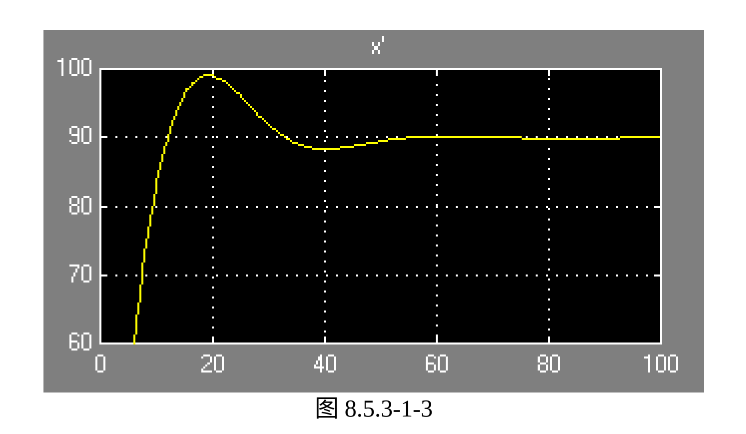


图 8.5.3-1-3

8.6 SIMULINK 的分析工具

8.6.1 确定模型的特征

【例 8.6.1-1】观察例 8.5.3-1 中所建模型 exm08053_1.mdl 中状态向量的结构。

```
[sizes,x0,StateCell]=exm08053_1;
SIZES=sizes',X0=x0',StateCell
SIZES =
     2     2     0     0     0     0     3
X0 =
     0     0     0     0
StateCell =
    'exm08053_1/Automobile Model/Int1'
    'exm08053_1/Automobile Model/Int2'
    'exm08053_1/PID Controller/DT-I'
    'exm08053_1/PID Controller/DD'
```

8.6.2 用 MATLAB 指令运行 SIMULINK 模型

8.6.2.1 运行 SIMULINK 模型的 sim 指令

8.6.2.2 设置编辑仿真参数的 simset 指令

8.6.2.3 获取模型仿真参数的 simget 指令

8.6.2.4 MATLAB 指令运行 SIMULINK 模型的示例

【例 8.6.2.4-1】以例 8.5.3-1 中所建模型 exm08053_1.mdl 为基础进行本题解算。演示：（A）显示模型窗中的初始状态设置。（B）把初始车速重置为 120，而其他初始值仍为 0。（B）画出两种初始状态下的车速曲线。

```
InInit=simget('exm08053_1','InitialState')      %获取模型窗对初始值的设置
[t,x,y]=sim('exm08053_1',100);                  %在模型内设置参数下进行仿真
opts=simset('InitialState',[120,0,0,0]);        %初始值的重置
[tt,xx,yy]=sim('exm08053_1',100,opts);          %在重置初值下仿真
plot(t,x(:,1),':b',tt,xx(:,1),'r')
legend('\fontname{隶书}\fontsize{16}内初值','外初值',4)
```

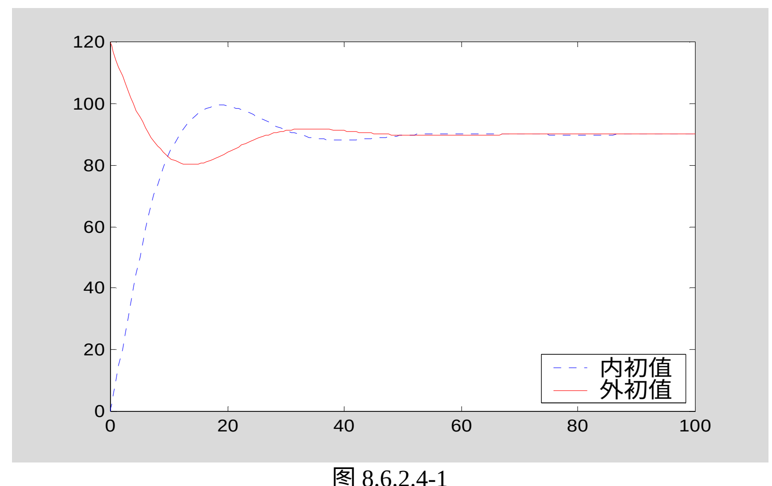


图 8.6.2.4-1

8.6.3 模型的线性化问题

8.6.3.1 线性化的数学描述

8.6.3.2 连续系统的线性化模型

8.6.3.3 离散系统的线性化模型

8.6.3.4 模型线性化的算例

【例 8.6.3.4-1】求非线性系统 $\begin{cases} \dot{x}_1 = x_1^{\ 2} + x_2^{\ 2} - 4 \\ \dot{x}_2 = 2x_1 - x_2 \end{cases}$ 在坐标原点处的线性化模型。

（1）
（2）

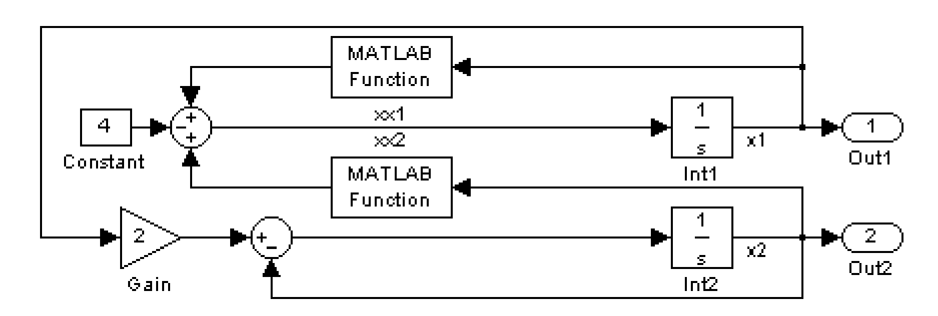


图 8.6.3.4-1

（3）

```
[A,B,C,D]=linmod('exm080634_1');A
A =
         0         0
    2.0000   -1.0000
```

（4）

```
[A1,B1,C1,D1]=linmod('exm080634_1',[1,0.5]);A1
A1 =
    2.0000    1.0000
```

```
    2.0000    -1.0000
```

（5）

```
eA=eig(A)',eA1=eig(A1)'
eA =
   -1.0000         0
eA1 =
    2.5616    -1.5616
```

8.6.4 系统平衡点的求取

8.6.5 综合算例

8.6.5.1 “ 一步仿真 ” 和精良状态轨迹斜率图

【例 8.6.5.1-1】求非线性系统 $\begin{cases} \dot{x}_1 = x_1^{\ 2} + x_2^{\ 2} - 4 \\ \dot{x}_2 = 2x_1 - x_2 \end{cases}$ 的相平面轨迹、平衡点，并进行稳定性分析。本例综合演示：（A）SIMULINK 模型和 MATLAB 指令的配合使用。（B）sim , simset , trim 指令的应用。（C） “ 一步仿真 ” 计算方法。（D）二阶系统相轨迹的精良图形。

（1）

（2）

[exm080651_1.m]

```
% exm080651_1.m
clf;hold on
xx=[-2,1;-1,1;0,1;1,1;1,0;1,-1;1,-2];
nxx=size(xx,1);
for k=1:nxx
   opts=simset('initialstate',[xx(k,1),xx(k,2)]);
   [t,x,y]=sim('exm080634_1',10,opts);
   plot(x(:,1),x(:,2));
end
xlabel('x1');ylabel('x2'),grid,hold off
```

（2）

```
exm080651_1
```

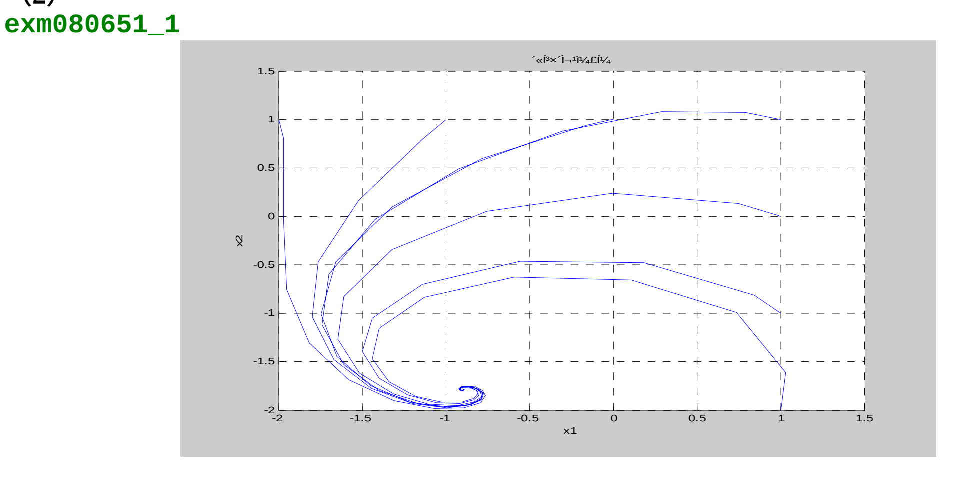


图 8.6.5.1-1-1

（4）

[portraitzzy.m]

```
function [DX1,DX2,DP]=portraitzzy(x1,x2,h)
% PORTRAITZZY
%
%
opts=simset('solver','ode5','fixedstep',h);     %   <7>
n=length(x1);
X1=zeros(n,n);X2=X1;
for ii=1:n;
   for jj=1:n;
      opts=simset(opts,'initialstate',[x1(ii),x2(jj)]);%  <12>
      [t,x,y]=sim('exm08634_1',h,opts);      %   <13>
      dx1=x(2,1)-x1(ii);
      dx2=x(2,2)-x2(jj);
      L=sqrt(dx1^2+dx2^2);。
      Z(jj,ii)=L;
      if L>1.e-10
         DX1(jj,ii)=dx1/L;DX2(jj,ii)=dx2/L;  %   <19>
      end
   end
end
DP=Z/h;
```

（5）

```
h=0.01;
x1=-2.5:0.25:2.5;x2=x1;
k=3.5;
[X1,X2]=portraitzzy(x1,x2,h);
quiver(x1,x2,k*X1,k*X2,0)
xlabel('x1'),ylabel('x2')
```

图 8.6.5.1-1-2

（6）

```
surfc(x1,x2,Z),view([18,32]),xlabel('x1'),ylabel('x2')
```

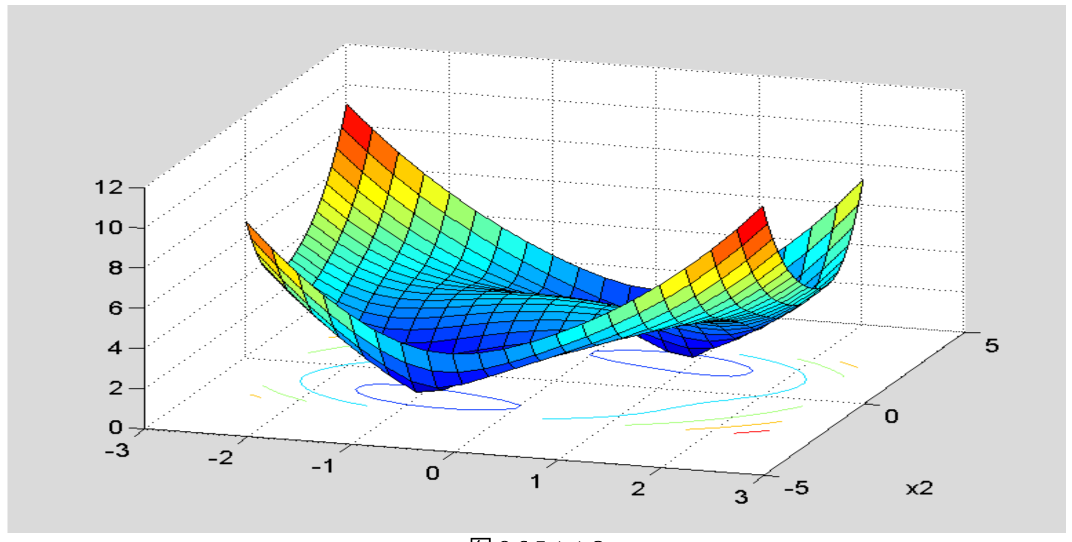


图 8.6.5.1-1-3

（6）

```
xa=trim('exm080634_1',[-1,-2]')
xb=trim('exm080634_1',[1,2]')
xa =
   -0.8944
   -1.7889
xb =
    0.8944
    1.7889
```

（7）

```
Axa=linmod2('exm080634_1',xa);eig_Axa=(eig(Axa))'
Axb=linmod2('exm080634_1',xb);eig_Axb=(eig(Axb))'
eig_Axa =
  -1.3944 - 2.6457i  -1.3944 + 2.6457i
eig_Axb =
    3.4110   -2.6222
```

8.6.5.2 仿真模型和优化指令的协调

【例 8.6.5.2-1】本例演示：（A）如何用 SIMULINK 模块计算性能函数。（B）SIMULINK 方块模型、目标函数和优化程序之间的协调和参数传递。（C）跨空间交换数据、跨空间计算表达式。

（1）

（2）

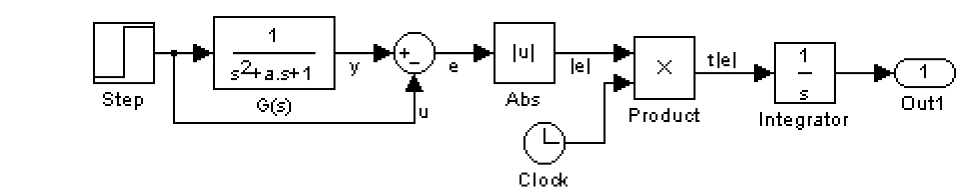


图 8.6.5.2-1-1

（3）

[itae.m]

```
function ss=itae(aa)
```

```
%
global a ss
a=aa;
Tspan=evalin('base','Tspan');      %  <5>
opts=simset('RelTol',0.0001);      %
[tt,x,s]=sim('smodel',Tspan,opts);%  <7>
ss=s(end);                         %  <8>
```

[exm080652_1.m]

[exm080652_1.m]

```
%exm08652_1.m
clear
global a ss
a0=[3.3 6.6 8.6 7.5 3.9];
Tspan=(0:500)/10;
options=optimset('LargeScale','off');
a=fminunc(@itae,a0,options);
coeff=[1,a,1]
ss
```

（4）

```
exm080652_1
coeff =
1.0000    2.1526    5.6292    6.9349    6.7927    3.7399    1.0000
ss =
    8.3338
```

（5）

```
old=tf(1,[1 3.25  6.60  8.60  7.45  3.95  1]);
new=tf(1,coeff);
[yold,told]=step(old,50);
[ynew,tnew]=step(new,50);
plot(told,yold,'b','LineWidth',1)
axis([3,18,0.95,1.05])
hold on,plot(tnew,ynew,'r','LineWidth',3),hold off
legend('Old','New',4),grid on
```

图 8.6.5.2-1-1

表 8.6.5.2-1 ITAE（单点）标准型新系数（黑体）和相应老“经典”系数对照

阶次	ITAE 值	传递函数分母多项式系数
2	1.99 **1.952**	1 1.4 1 **1 1.5 1**

3	3.144	1 1.75 2.15 1
	3.138	**1 1.78 2.17 1**
4	4.626	1 2.10 3.40 2.75 1
	4.591	**1 1.95 3.35 2.65 1**
5	7.155	1 2.80 5.00 5.50 3.40 1
	6.319	**1 2.07 4.49 4.67 3.26 1**
6	9.656	1 3.25 6.60 8.60 7.45 3.95 1
	8.333	**1 2.17 5.63 6.96 6.79 3.74 1**
7	15.003	1 4.48 10.42 15.05 15.54 10.64 4.580 1
	10.628	**1 2.22 6.750 9.360 11.58 8.680 4.320 1**
8	18.680	1 5.20 12.80 21.60 25.75 22.20 13.30 5.15 1
	13.222	**1 2.53 8.020 12.72 17.98 16.70 11.49 4.86 1**

表 8.6.5.2-2 ITAE（区域）标准型系数

阶次	**ITAE 均值**	**传递函数分母多项式系数**
2	1.999	1 1.5±0.2 1
3	3.241	1 1.82±0.15 2.19±0.12 1
4	5.597	1 2.32±0.41 3.60±0.31 2.80±0.24 1
5	10.74	1 4.25±0.46 6.48±0.78 6.85±0.75 3.84±0.32 1
6	11.99	1 2.77±0.35 6.91±0.49 8.73±0.65 8.14±0.61 4.09±0.41 1
7	17.86	1 2.54±0.34 9.13±0.25 14.97±0.35 17.31±0.45 12.36±0.30 5.24±0.53 1
8	19.92	1 3.77±0.14 8.35±0.12 16.15±0.15 17.69±0.09 18.34±0.18 10.74±0.14 4.77±0.65 1

8.7 数值计算方面的考虑

8.7.1 微分方程解算器 Solver

8.7.1.1 ODE45 和 ODE23 运作机理简要

8.7.1.2 ODE113 运作机理简要

8.7.1.3 ODE15S 和 ODE23S 运作机理简要

8.7.1.4 不同解算器处理 Stiff 系统时表现

【例 8.7.1.4-1】求微分方程 $\ddot{x}+100\dot{x}+0.9999x=0$ 在 $x(0)=1,\dot{x}(0)=0$ 时的解。本例演示：对于 Stiff 方程，如果解算方法选择不当将产生严重后果。

（1）

```
xsym=dsolve('D2x+100*Dx+0.9999*x=0','x(0)=1,Dx(0)=0','t')
dxsym=diff(xsym,'t')
xsym =
9999/9998*exp(-1/100*t)-1/9998*exp(-9999/100*t)
dxsym =
-9999/999800*exp(-1/100*t)+9999/999800*exp(-9999/100*t)
```

（2）

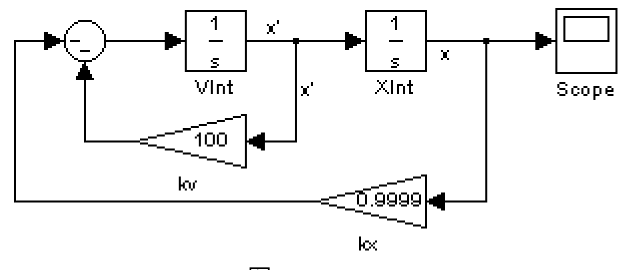


图 8.7.1.4-1-1

(3)

```
t=(0:5000)/10;x=eval(char(dxsym));
Tspan=500;
opts=simset('Solver','ode45');
[tt1,xx1,s]=sim('exm080714_1',Tspan,opts);
opts=simset('Solver','ode15s');
[tt2,xx2,s]=sim('exm080714_1',Tspan,opts);
plot(t,x,'k',tt1,xx1(:,2),'b:',tt2,xx2(:,2),'r-.')
axis([246 247 -8.55e-4  -8.35e-4])
legend('Symbolic','ODE45','ODE15S',0)
ns1=length(xx1)
ns2=length(xx2)
ns1 =
    15072
ns2 =
    101
```

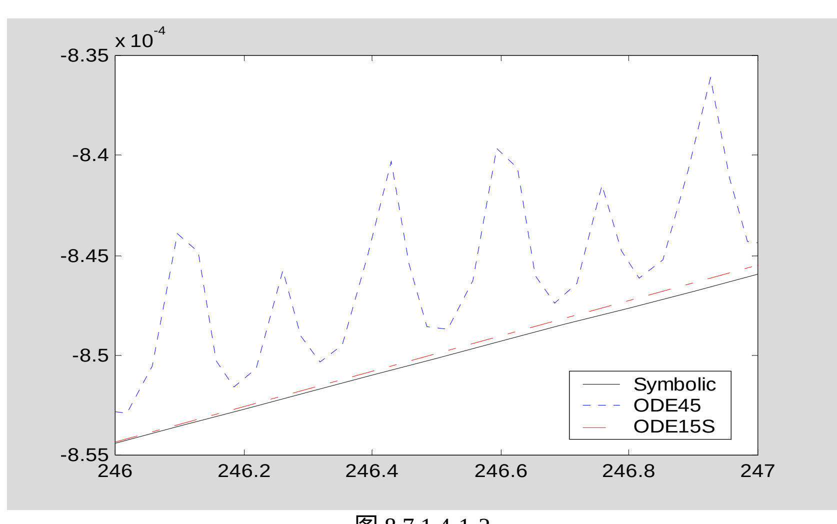


图 8.7.1.4-1-2

8.7.2 积分步长和容差

8.7.2.1 积分步长的选择

【例 8.7.2.1-1】以例 8.5.2-1 的多采样模型 exm08052_1.mdl 为基础。试验解算器、工作模式、采样转移模块的影响。

8.7.2.2 计算容差的选择

8.7.3 代数环问题

8.7.3.1 代数环的形成

【例 8.7.3.1-1】研究方程组 $\dot{x} = u - k_3 y, y = k_1 x + k_2 \dot{x}$的解算问题。

（1）

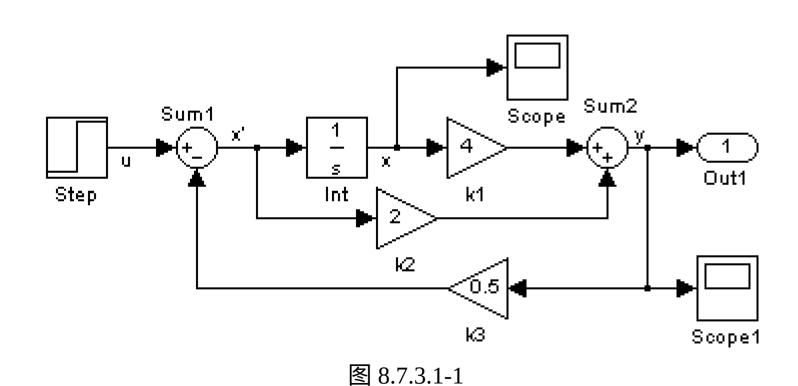


图 8.7.3.1-1

8.7.3.2 代数环的处理

【例 8.7.3.2-1】本例以例 8.7.3.1-1 为基础。演示：通过重组模型，直接消除代数环，建立等价模型。

（1）

（2）

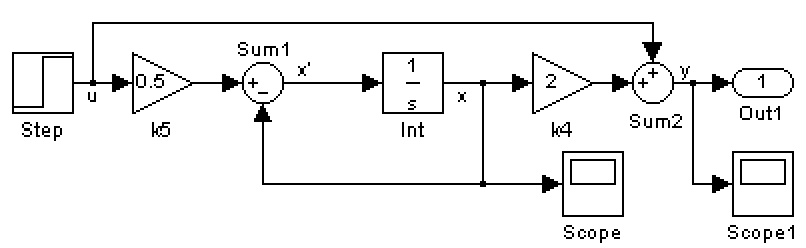


图 8.7.3.2-1

【例 8.7.3.2-2】本例以例 8.7.3.1-1 为基础。演示：（A）利用“ 记忆 ”模块中断代数环。（B）“ 记忆 ” 模块的副作用。（C）与前两个模型做性能比较。

（1）

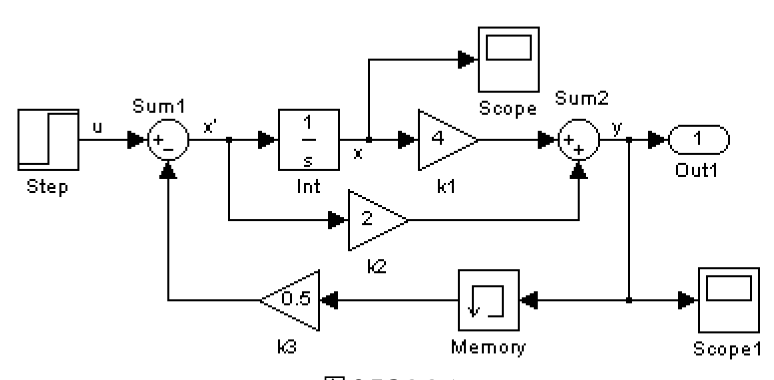


图 8.7.3.2-2-1

（2）

```
clear all
tic;[t1,x1,y1]=sim('exm080731_1',5);T1=toc;
tic;[t2,x2,y2]=sim('exm080732_1',5);T2=toc;
tic;[t3,x3,y3]=sim('exm080732_2',5);T3=toc;
disp([blanks(4),'有代数环',blanks(3),'无代数环',blanks(3),'带记忆块'])
```

```
disp([T1,T2,T3])
plot(t1,y1,'g','LineWidth',6),hold on
plot(t2,y2,'r','LineWidth',2)
plot(t3,y3,'b'),hold off
legend('with loop','without loop','memory',4)
Warning: Block diagram 'exm080731_1' contains 1 algebraic loop(s).
Found algebraic loop containing block(s):
  'exm080731_1/k3'
  'exm080731_1/Sum'
  'exm080731_1/k2'
  'exm080731_1/Sum1' (algebraic variable)
    有代数环    无代数环    带记忆块
    0.2800     0.2200     0.1100
```

图 8.7.3.2-2-2

8.8 S 函数模块

8.8.1 概述

8.8.2 SIMULINK 的运行机理

8.8.3 用 M 文件表述的 S 函数

8.8.3.1 S 函数的模板程序

8.8.3.2 S 函数模块的创建和使用

【例 8.8.3.2-1】本例演示：（1）用 S 函数模块为图 8.8.3.2-1 所示单摆构造系统动力学模型；（2）利用 SIMULINK 研究该单摆摆角 θ 的运动曲线；（3）用 S 函数动画模块表现单摆的运动。

（1）

（2）

（3）

[simpendzzy.m]

```
function [sys,x0,str,ts] = simpendzzy(t,x,u,flag,dampzzy,gravzzy,angzzy)
%
%
```

```
%
%
switch flag,
  case 0,
    [sys,x0,str,ts]=mdlInitializeSizes(angzzy);
  case 1,
    sys=mdlDerivatives(t,x,u,dampzzy,gravzzy);
  case 2,
    sys=mdlUpdate(t,x,u);
  case 3,
    sys=mdlOutputs(t,x,u);
  case 9,
    sys=mdlTerminate(t,x,u);
  otherwise
    error(['Unhandled flag = ',num2str(flag)]);
end

% ------ mdlInitializeSizes ------------------------
function [sys,x0,str,ts]=mdlInitializeSizes(angzzy)
sizes = simsizes;
sizes.NumContStates  = 2;
sizes.NumDiscStates  = 0;
sizes.NumOutputs     = 1;
sizes.NumInputs      = 1;
sizes.DirFeedthrough = 0;
sizes.NumSampleTimes = 1;
sys = simsizes(sizes);
x0  = angzzy;
str = [ ];
ts  = [0, 0];
% -----mdlDerivatives ----------------------------
function sys=mdlDerivatives(t,x,u,dampzzy,gravzzy)
dx(1)=-dampzzy*x(1)-gravzzy*sin(x(2))+u;
dx(2)=x(1);
sys =dx ;
% -----mdlUpdate ---------------------------------
function sys=mdlUpdate(t,x,u)
sys = [ ];
% -----mdlOutputs --------------------------------
function sys=mdlOutputs(t,x,u)
sys = x(2);
% -----mdlTerminate ------------------------------
function sys=mdlTerminate(t,x,u)
sys = [ ];
```

（4）

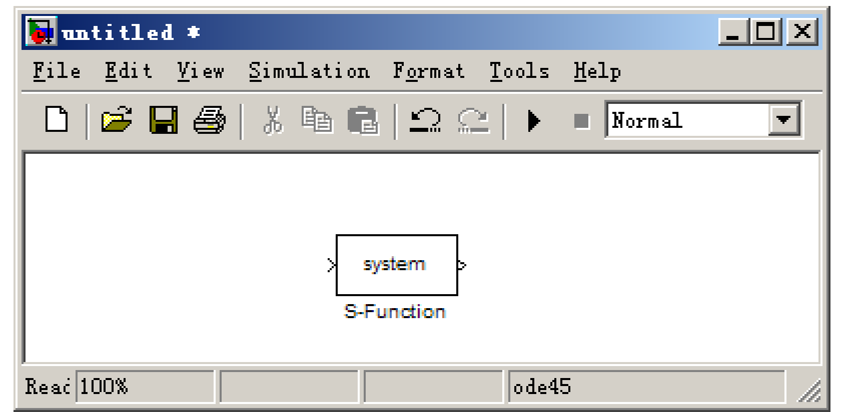


图 8.8.3.2-2

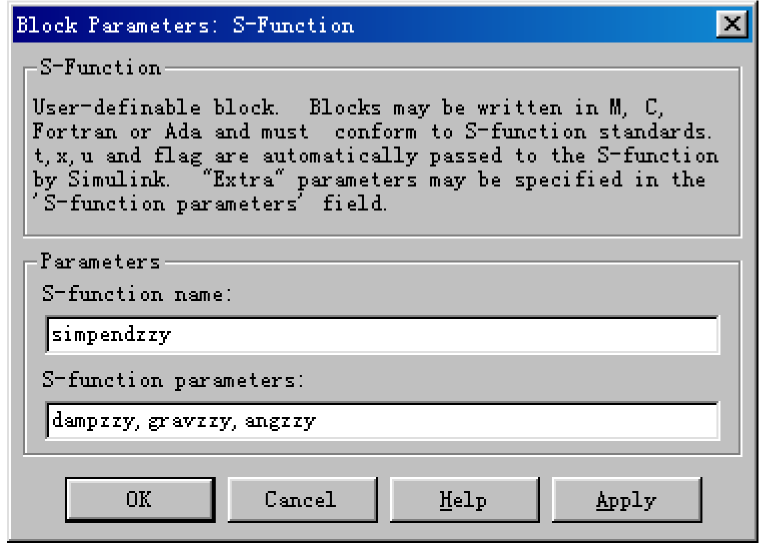


图 8.8.3.2-3

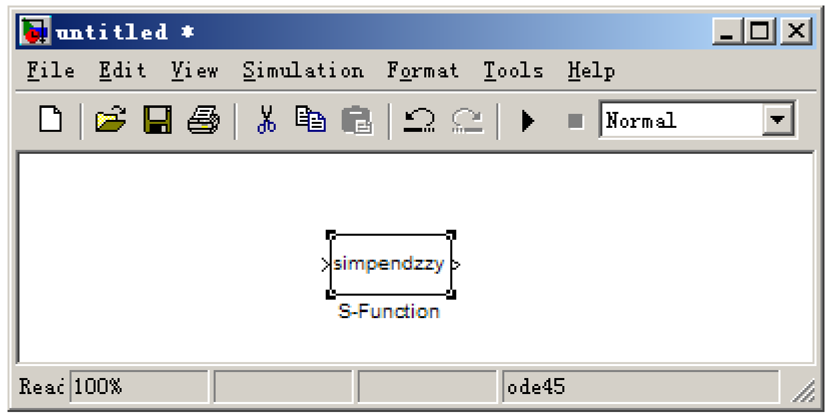


图 8.8.3.2-4

（5）

```
clear
dampzzy=0.8;gravzzy=2.45;angzzy=[0;0];
```

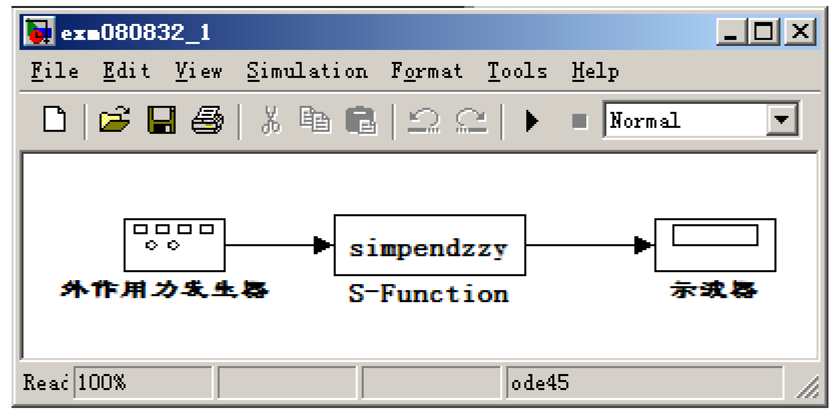


图 8.8.3.2-5

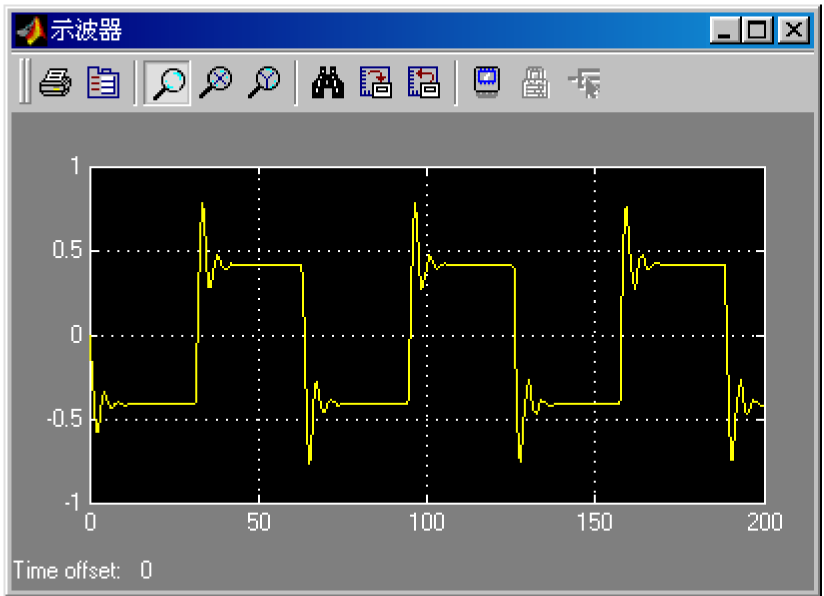


图 8.8.3.2-6

(6)

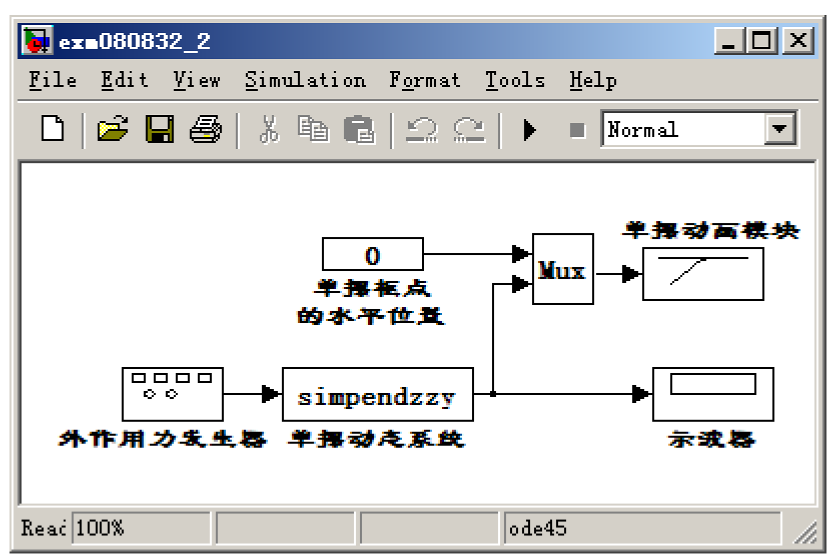
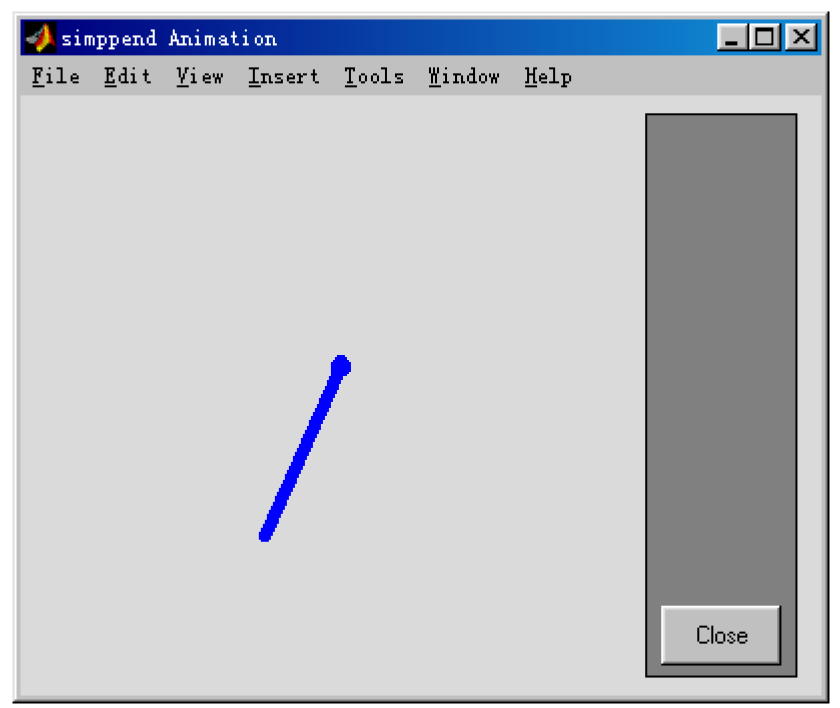


图 8.8.3.2-7


simppend Animation
File
Edit
View
Insert
Tools
Window
Help
Close
图 8.8.3.2-8

第九章　句柄图形

与第 6 章的高层指令相比，本章的内容更深入 MATLAB 可视化功能的内核。编写本章的目的有两个：一，使读者更深入地理解高层绘图指令，从而可绘制出更精细更生动更个性的图形；二，使读者能利用低层图形指令和图形对象属性开发专用绘图函数。

本章在内容安排上有如下考虑：

- **保证概念、结构和方法的完整性**：本章的前 6 节内容按由表及里、由浅入深的原则系统阐述句柄图形体系、图形对象、属性和操作方法。
- **突出要点、新点和难点**：句柄图形体系有 11 个基本图形对象，每个对象的属性少则 20 几个，多则近百个。对此，MATLAB 自带资料有详尽的文字说明。因此，本章只对最常用的、不可或缺的、以及较难掌握的内容进行说明。
- **强调“可操作性”体现“范例引导概念”的本书宗旨**：针对 MATLAB 自带资料缺少完整、可操作应用实例的弱点，本章设计了 17 个算例，其中 9 个精心设计的完整应用范例就占本章一半以上篇幅。读者通过阅读或操作这些范例，可掌握各指令、属性之间的有机配合，从而更具体更真切地理解句柄图形。

在本章内容正式展开之前，先把 MATLAB 随带资料中涉及各对象属性的文件及查阅方法罗列如下。读者应重视这些最原始、最权威、最细节、任何其它书籍不能代替的资料。

- 使用 Adobe Acrobat Reader 阅读（或打印）以下 PDF 文件
  help\pdf_doc\matlab\graphg.pdf ；
  help\pdf_doc\matlab\refbook.pdf, refbook2.pdf, refbook3.pdf ；
- 使用 Help Navigator/Browser 帮助导航/浏览器或直接指令 help 查看有关图形对象的资料。如 help figure 可直接得到有关图对象的属性描述。
- 利用 get , set 指令在 MATLAB 指令窗中，直接查询对象属性。

本章内容已根据 MATLAB6.5 版中运行情况修正。但值得指出：MATLAB 从 5.x 版向 6.x 版的升级对本章内容的扩展和影响很小。

9.1 句柄图形体系

9.1.1 图形对象、对象句柄和句柄图形树结构

（1）图形对象
（2）句柄
（3）句柄图形的结构

9.1.2 对象属性

（1）属性
（2）缺省属性

9.2 图形对象的操作

9.2.1 图形对象创建指令一览

9.2.2 对象句柄的获取方法

9.2.2.1 基本方法

（1）从图形创建指令获得句柄
（2）追溯法获取图柄
（3）当前对象句柄的获取
（4）根据对象特性获取句柄
（5）根据对象“ 标签 ”获取句柄

9.2.2.2 句柄获取示例

【例 9.2.2.2-1】画网线图，并得相应句柄；追溯法找所在图形窗句柄；gcf 和 gca 演示。（为省篇幅，图形略）

```
clf reset;H_mesh=mesh(peaks(20))
H_grand_parent=get(get(H_mesh,'Parent'),'Parent')
disp('      图柄      轴柄'),disp([gcf gca])
H_mesh =
  100.0016
H_grand_parent =
     1
      图柄      轴柄
    1.0000   99.0010
```

【例 9.2.2.2-2】低层指令绘图，获得句柄；获取同轴上字对象的句柄和相应对象类型。（为省篇幅，图形略）

```
clf reset,t=(0:100)/100*2*pi;H_line=line('Xdata',t,'Ydata',sin(t))
text(pi,0.8,'\fontsize{14}sin(t)')
H_c=get(get(H_line,'parent'),'children')
T=get(H_c,'Type')
H_line =
   99.0011
H_c =
  101.0005
   99.0011
T =
    'text'
    'line'
```

【例 9.2.2.2-3】findobj 指令的使用。（为省篇幅，图形略）

```
clf reset,t=(0:pi/100:2*pi)';tt=t*[1 1];yy=sin(tt)*diag([0.5 1]);
plot(tt,yy),Hb=findobj(gca,'Color','b')
Hb =
   99.0012
```

9.3 对象属性的获取和设置

9.3.1 创建对象时设置属性

9.3.2 get 和 set

9.3.3 对象属性的缺省设置和查询

9.3.4 属性查询和设置示例

【例 9.3.4-1】创建二维图形时，分别用元胞数组和构架数组设置对象属性。

```
clf reset,x=0:pi/12:2*pi;
PN1={'Color', 'LineWidth','Marker'};
PV1={[1 0 0], 5 , 'd'};
plot(sin(x),cos(x),PN1,PV1)
axis square off
PS.Color=[0.7 0.7 0];PS.LineWidth=2;
line(sin(7*x),cos(7*x),PS);
```

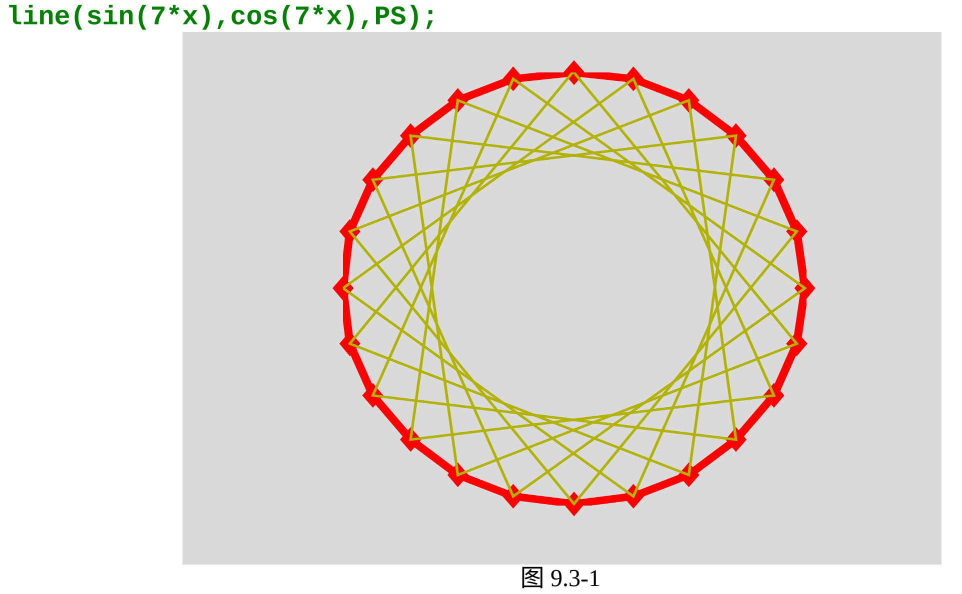


图 9.3-1

【例 9.3.4-2】本例演示：影响 line 或 plot 画线时线型和色彩的“父”对象设置。出于篇幅考虑，本节算例的图形都被删去。如要观察指令产生的图形，请读者自己运行相应指令。

(1)

```
t=(0:pi/50:2*pi)';k=0.4:0.1:1;Y=cos(t)*k;
line(t,Y,'Color',[1 0 0],'LineStyle','-.')                    %<2>
```

(2)

```
clf reset
set(gca,'ColorOrder',[0 0 0;0.7 0.7 0.7],'LineStyle','-|:')    %<4>
line(t,Y)
```

(3)

```
clf reset
set(gcf,'DefaultAxesLineStyleOrder','-|:');                    %<8>
set(gcf,'DefaultAxesColorOrder',[1 0 0;0 0 1]);
line(t,Y)
```

9.4 为低层指令绘图准备图/轴

9.4.1 'NextPlot'属性

9.4.2 准备图/轴的简捷指令 newplot

9.4.3 高层绘图文件的形成

【例 9.4.3-1】高层作图函数 surf.m 文件与底层作图指令 **surface** 的关系。

[surf.m]

```
function h = surf(varargin)
cax = newplot;
if nargin == 0
    error('Not enough input arguments.')
elseif nargin == 1
    if min( size( varargin{1} ) ) == 1
        error('Input argument must be a matrix not a vector or a scalar')
    else
        hh = surface(varargin{1});
    end
else
    hh = surface(varargin{:});
end
next = lower(get(cax,'NextPlot'));
if ~ishold
    view(3)
    grid on
end
if nargout == 1
    h = hh;
end
```

9.5 图形窗的色彩资源和光标属性

9.5.1 色彩资源

9.5.2 光标指针

9.5.2.1 预定义的指针形状

9.5.2.2 自定义指针形状

9.6 轴对象

9.6.1 轴位框的几何属性和多轴位框

9.6.2 图形名和坐标轴名的句柄操作

9.6.3 轴刻度的属性控制

9.6.4 坐标轴尺度、方向、位置属性

9.6.5 照相机属性

9.7 句柄图形应用专题

9.7.1 光标形状的自制

【例 9.7.1-1】自制光标指针形状

（1）

```
bdw=0.01;
tpw=0.15;
pos=[1/2+bdw,2/3+bdw,1/2-2*bdw,1/3-bdw-tpw];                   %<3>
figure('Units','normalized','Position',pos,'Color',[0.9,0.65,0])
                                                            %<4>
set(gcf,'Name','试验窗')                                     %<5>
```

（2）

步骤一：

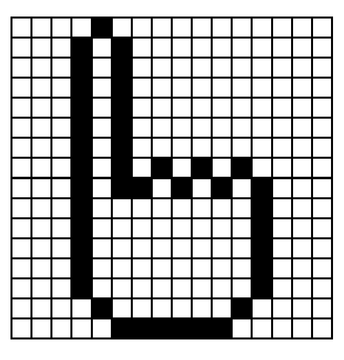


图 9.7-1

步骤二：

```
P=ones(16,16)*NaN;
P(1,5)=1;P(2:14,4)=1;P(15,5)=1;P(16,6:11)=1;P(15,12)=1;P(9:14,13)=1;
P(2:9,6)=1;P(9,7)=1;P(8,8)=1;P(9,9)=1;P(8,10)=1;P(9,11)=1;P(8,12)=1;
P(2:14,5)=2;P(10:15,6:11)=2;P(9,8)=2;P(9,10)=2;P(9:14,12)=2;
set(gcf,'Pointer','Custom','PointerShapeCData',P,'PointerShapeHotSpot
',[2,5])
```

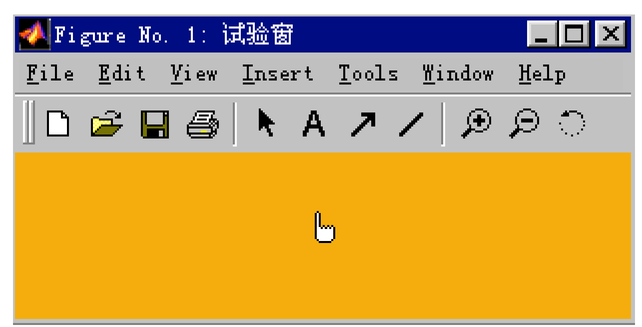


图 9.7-2

9.7.2 任意布置子图和轴外注释

【例 9.7.2-1】本例演示：axes 轴位框设计、rectangle 的运用、及轴外注释。所谓轴外注释，实际上是使用了两个轴位框。一个轴位框充满全部图形窗，其坐标框被隐去，而只写注释文字。而另一个比较小的轴位框用于绘图。这样从外表看去，注释就处于那小轴位框的外部。

```
clf
%
zeta2=[0.2 0.4 0.6 0.8 1.0];n=length(zeta2);
for k=1:n;Num{k,1}=1;Den{k,1}=[1 2*zeta2(k) 1];end
S=tf(Num,Den);
t=(0:0.4:20)';
[Y,x]=step(S,t);
tt=t*ones(size(zeta2));
%
clf reset,H=axes('Position', [0, 0, 1, 1], 'Visible', 'off');
%
str{1}='\fontname{隶书}二阶系统阶跃响应';                            %<11>
str{2}='y(t) = 1 - \beta^{-1}e^{-\zetat}sin(\betat + \theta)';
str{3}=' ';str{4}='\fontname{隶书}其中：';
str{5}='\beta = (1 - \zeta^{2})^{0.5}';
str{6}='\theta = arctg(\beta/\zeta)';
str{7}='\zeta = .2, .4, .6, .8, 1';                                  %<15>
%
set(gcf, 'CurrentAxes',H)                                            %<18>
text(0.01, 0.73, str, 'FontSize', 12)                                %<19>
h1=axes('Position', [0.45, 0.45, 0.5, 0.5]);
ribbon(tt,Y,0.4)
%
set(h1,'XTickLabelMode','manual','XTickLabel','0|0.4|0.8|1.2');%<23>
set(h1,'ZTickLabel','0|1.0|2.0');                                    %<24>
%
set(get(h1,'XLabel'),'String','\zeta \rightarrow','Rotation',17.5)
set(get(h1,'YLabel'),'String','\leftarrow t','Rotation',-25)   %<27>
set(get(h1,'Zlabel'),'String','y \rightarrow')
h2=axes('Position',[0.03, 0.08, 0.27, 0.27]);%
plot(tt,Y)                                     %
%
h3=axes('Position',[0.37,0.04,0.63,0.32]);   %
set(h3,'Xlim',[0,1.2],'Ylim',[0,0.5])        %
set(h3,'DataAspectRatio',[1 1 1])            %
set(h3,'ColorOrder',[0,0,0])                 %
set(h3,'Visible','off')                      %
hh1=rectangle('Position',[0.5,0.2,0.4,0.2],'Curvature',[0,0]);
                                             %  <37>
hh2=rectangle('Position',[0.2,0.26,0.08,0.08],'Curvature',[1,1]);
                                             %  <38>
xx1=0.05:0.01:0.2;xx2=0.28:0.02:0.5;
xx3=0.9:0.02:1.1;xx4=0.24:0.02:1;
yy5=0.1:0.02:0.26;yy6=0.1:0.02:0.3;
yy1=0.3*ones(size(xx1));yy2=0.3*ones(size(xx2));
yy3=0.3*ones(size(xx3));yy4=0.1*ones(size(xx4));
xx5=0.24*ones(size(yy5));xx6=ones(size(yy6));
line(xx1,yy1);line(xx2,yy2);line(xx3,yy3);line(xx4,yy4);
line(xx5,yy5);line(xx6,yy6)
line(0.17,0.3,'Marker','>','MarkerFaceColor','k')
line(0.47,0.3,'Marker','>','MarkerFaceColor','k')
line(1.1,0.3,'Marker','>','MarkerFaceColor','k')
```

```
line(0.24,0.23,'Marker','^','MarkerFaceColor','k')
line(0.17,0.35,'Marker','+')
text(0.27,0.23,'-')
text(0.05,0.35,'u(t)')
text(1,0.35,'y(t)')
text(0.6,0.26,'s{^2} + 2{\zeta}s');
xx7=0.56:0.02:0.84;yy7=0.3*ones(size(xx7));line(xx7,yy7)
text(0.68,0.35,'1')
```

图 9.7-3

9.7.3 制作个性化双坐标系

【例 9.7.3-1】制作一个双坐标系用来表现高压和低温两个不同量的过渡过程。

```
tp=(0:100)/100*5;yp=8+4*(1-exp(-0.8*tp).*cos(3*tp));
tt=(0:500)/500*40;yt=120+40*(1-exp(-0.05*tt).*cos(tt));
%
clf reset,h_ap=axes('Position',[0.13,0.13,0.7,0.75]);          %<4>
set(h_ap,'Xcolor','b','Ycolor','b','Xlim',[0,5],'Ylim',[0,15]);
nx=10;ny=6;                                                     %<6>
pxtick=0:((5-0)/nx):5;pytick=0:((15-0)/ny):15;                 %<7>
set(h_ap,'Xtick',pxtick,'Ytick',pytick,'Xgrid','on','Ygrid','on')
h_linet=line(tp,yp,'Color','b');                               %<9>
set(get(h_ap,'Xlabel'),'String','时间 \rightarrow （分）')
set(get(h_ap,'Ylabel'),'String','压力 \rightarrow(\times10 ^{5} Pa )')
h_at=axes('Position',get(h_ap,'Position'));                  %<12>
set(h_at,'Color','none','Xcolor','r','Ycolor','r');            %<13>
set(h_at,'Xaxislocation','top')                                %<14>
set(h_at,'Yaxislocation','right','Ydir','rev')                 %<15>
set(get(h_at,'Xlabel'),'String','\fontsize{15}\fontname{ 隶 书 } 时 间
\rightarrow （分）')
set(get(h_at,'Ylabel'),'String',' ( {\circ}C )\fontsize{15} \leftarrow
\fontname{隶书}零下温度')
```

```
set(h_at,'Ylim',[0,210])                                    %<18>
line(tt,yt,'Color','r','Parent',h_at)                       %<19>
xpm=get(h_at,'Xlim');                                       %<20>
txtick=xpm(1):((xpm(2)-xpm(1))/nx):xpm(2);                  %<21>
tytick=0:((210-0)/ny):210;                                  %<22>
set(h_at,'Xtick',txtick,'Ytick',tytick)                     %<23>
```

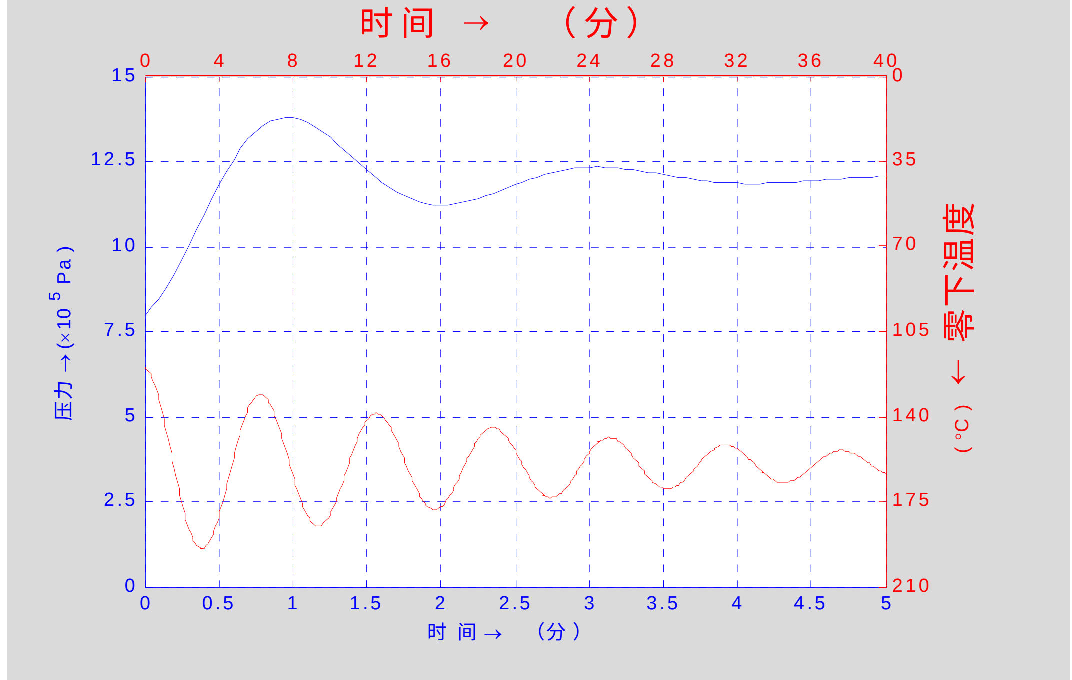


图 9.7-4

9.7.4 连续变焦和飞驰图形

【例 9.7.4-1】通过 CameraPosition 设置的不断变化，使地球迎面飞来，贯穿而过，从地球另一面飞离而去。但在整个飞行过程中，相机镜头始终对着地球。

（1）

[earth_zzy.m]

```
function earth_zzy(ap)
%erath_zyy.m
%
%
load topo
figure('colormap',topomap1,'Color',[.8 .8 .8]); %  <6>
[x,y,z] = sphere(50);
azzy.DataAspectRatio = [1 1 1];azzy.PlotBoxAspectRatioMode = 'auto';
fa = axes('Visible','off', azzy);
szzy1.AmbientStrength = 0.1;szzy1.DiffuseStrength = 1;
szzy1.SpecularColorReflectance = .5; szzy1.SpecularExponent = 20;
szzy1.SpecularStrength = 1;
surface(x,y,z,szzy1,'FaceLighting','phong','FaceColor','texture',...
       'EdgeColor','none','Cdata',topo,'Parent',fa);% <13>
if ap==1,set(fa,'CameraViewAngle',0.1*get(fa,'CameraViewAngle'));end
light('position',[-1 0 1],'color',[0.5 1 0.5]);
light('position',[-1.5 0.5 -0.5],'color',[.6 .2 .2]);
light('Position',[1.5 1.5 -1]);
```

```
light('Position',[0 -1.5 0],'color',[0.6 0.6 1]);
view([-17 26])
```

（2）

```
earth_zzy(0)
```

图 9.7-5

（3）

[fly_zzy.m]

```
%fly_zzy.m
earth_zzy(0)                                              %   <2>
set(gca,'CameraViewAngleMode','manual')                   %   <3>
pos=get(gca,'CameraPosition');
tar=get(gca,'CameraTarget');
kk=(0:2:40)/15;nk=length(kk);                             %   <6>
for i=1:nk-1
   newpos=pos-kk(i)*(pos-tar);              %   <8>
   set(gca,'CameraPosition',newpos)         %   <9>
   drawnow                                  %   <10>
end
```

（4）

```
fly_zzy
```

【例 9.7.4-2】利用属性 CameraViewAngle 产生飞驰效应。本例也用地球图形演示。在相机视角连续变化下，地球飞离，直到消失。

[fly_zzy2.m]

```
%fly_zzy2.m
earth_zzy(1)
set(gca,'CameraViewAngleMode','manual')
ang=get(gca,'CameraViewAngle');
kk=(1:50)/50;nk=length(kk);
for i=1:nk
   newang=ang+kk(i)*(180-ang);
   set(gca,'CameraViewAngle',newang)
   drawnow
end
```

9.7.5 实时动画

9.7.5.1 擦除属性 'EraseMode'

9.7.5.2 屏幕刷新指令 drawnow

9.7.5.3 动画制作示例

【例 9.7.5.3-1】制作红色小球沿一条带封闭路径的下旋螺线运动的实时动画。

（1）

[anim_zzy1.m]

```
function f=anim_zzy1(K,ki)
% anim_zzy1.m
%
%
t1=(0:1000)/1000*10*pi;x1=cos(t1);y1=sin(t1);z1=-t1;
t2=(0:10)/10;x2=x1(end)*(1-t2);y2=y1(end)*(1-t2);z2=z1(end)*ones(size(x2));
t3=t2;z3=(1-t3)*z1(end);x3=zeros(size(z3));y3=x3;
t4=t2;x4=t4;y4=zeros(size(x4));z4=y4;
x=[x1 x2 x3 x4];y=[y1 y2 y3 y4];z=[z1 z2 z3 z4];
plot3(x,y,z,'b'),axis off
%
h=line('Color',[1 0 0],'Marker','.','MarkerSize',40,'EraseMode','xor');
%
n=length(x);i=1;j=1;
while 1
    set(h,'xdata',x(i),'ydata',y(i),'zdata',z(i));
   drawnow;                                        %   <21>
   pause(0.0005)                                   %   <22>
   i=i+1;
   if nargin==2 & nargout==1
      if(i==ki&j==1);f=getframe(gcf);end           %   <25>
   end
   if i>n
      i=1;j=j+1;
      if j>K;break;end
   end
end
```

（2）

```
f=anim_zzy1(2,450);
```

（3）

```
image(f.cdata),axis off
```

图 9.7-6

9.7.6 surface 指令衍生不同曲面

【例 9.7.6-1】演示高层指令 mesh, surf 等指令是如何由 surface 衍生而得。

```
clf reset,t=(0:20)/20;r=2.5-cos(2*pi*t);[x,y,z]=cylinder(r,40);
fc = get(gca,'color');
h=surface(x,y,z,'FaceColor',fc,'EdgeColor','flat','FaceLighting',
'none','EdgeLighting','flat');%  <4>
view(3); grid on;axis off
```

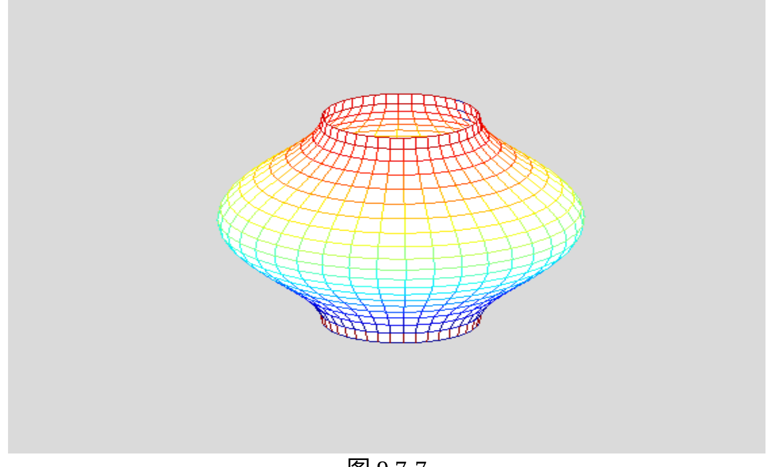


图 9.7-7

```
%
set(h,'FaceColor','flat','LineStyle','-','EdgeColor',[.8 .8 .8]) %<6>
```

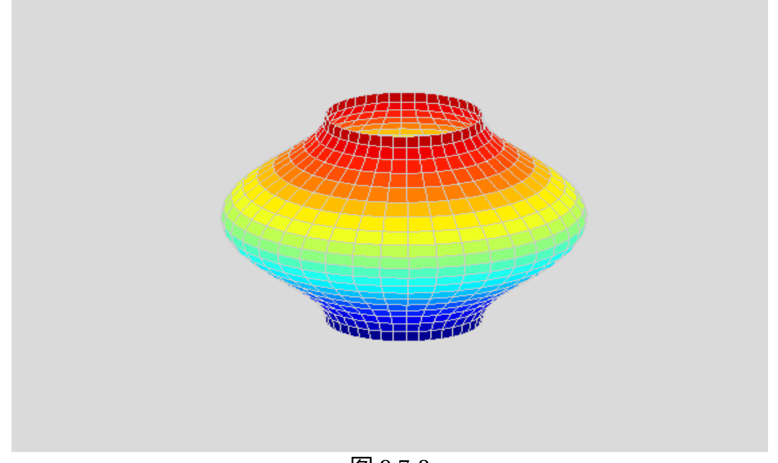


图 9.7-8

```
%
set(h,'FaceColor','interp','MeshStyle','column')                %<7>
```

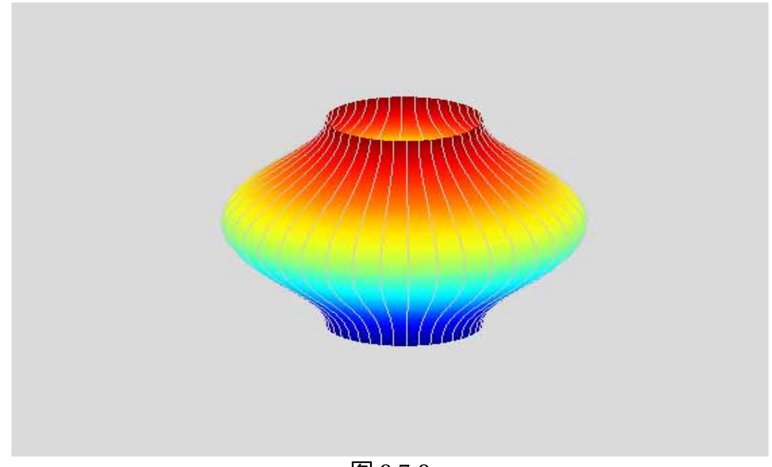


图 9.7-9

9.7.7 纹理影射和曲面彩绘

【例 9.7.7.-1】在曲面上彩绘 **unit8** 编址图象

```
clf reset,t=(0:20)/20;r=2.5-cos(2*pi*t);
[x,y,z]=cylinder(r,40);
%
[C,CMAP]=imread('trees.tif');
CC=double(C)+1;
%
surface(x,y,z,'Cdata',flipud(CC),'FaceColor','texturemap','EdgeColor'
,'none','CDataMapping','direct','Ambient',0.6,'diffuse',0.8,'specular
s',0.9)
colormap(CMAP)
view(3),axis off
```

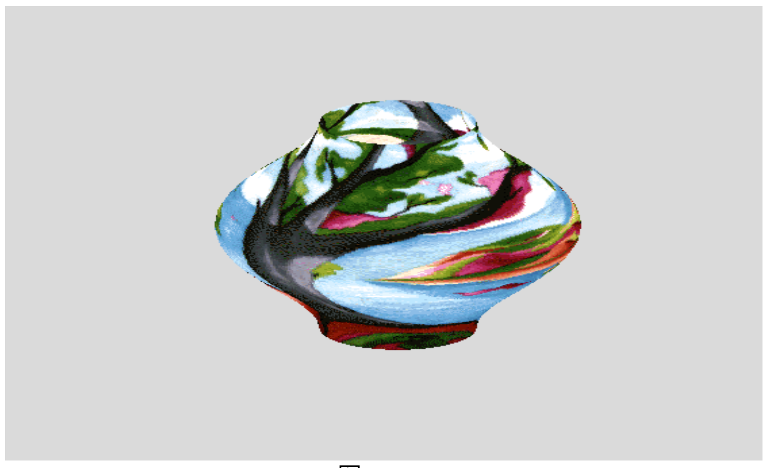


图 9.7-10

9.7.8 三维块建模和着色

9.7.8.1 创建块对象的顶点坐标法

【例 9.7.8.1-1】顶点坐标法创建三维长方块。

```
clf reset,k=8;
X=[0 1 1 0;1 1 1 1;1 0 0 1;0 0 0 0;0 1 1 0;0 1 1 0]';
```

```
Y=5*[0 0 0 0;0 1 1 0;1 1 1 1;1 0 0 1;0 0 1 1;0 0 1 1]';
Z=[0 0 1 1;0 0 1 1;0 0 1 1;0 0 1 1;0 0 0 0;1 1 1 1]';
FC=k:(k+size(Z,2)-1);
patch(X,Y,Z,FC),set(gca,'Projection','pers')
view([-20 -12]),colormap(jet),axis equal off
```

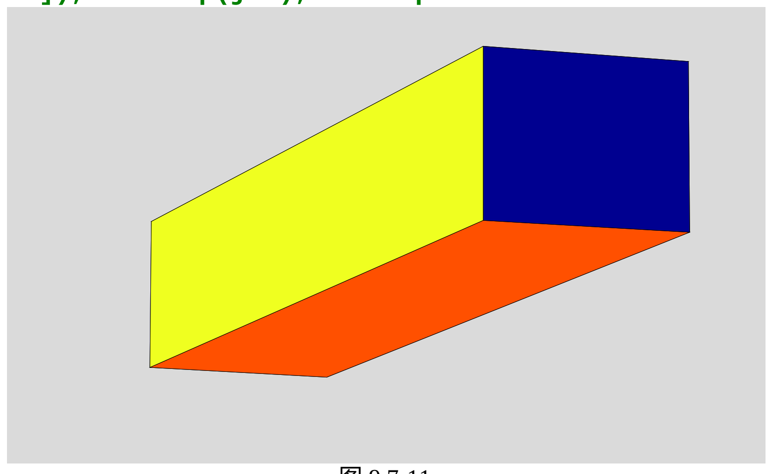


图 9.7-11

9.7.8.2 创建块对象的“ 顶/面 ”法

【例 9.7.8.2-1】“ 顶/面 ”法创建块对象。

```
clf reset
VM=[0 0 0;1 0 0;1 1 0;0 1 0;0 0 1;1 0 1;1 1 1;0 1 1;1/2  1/2  1+sqrt(2)/2];
FM=[1 2 6 5;2 3 7 6;3 4 8 7;4 1 5 8;5 6 9 5;6 7 9 6;7 8 9 7];
                                                             %<3>
VC=jet(9);FC='interp';
patch('Vertices',VM,'Faces',FM,'FaceVertexCData',VC,'FaceColor',FC)
set(gca,'Visible', 'off','DataAspectRatio',[1 1 1])         %<6>
view([-127 62])
```

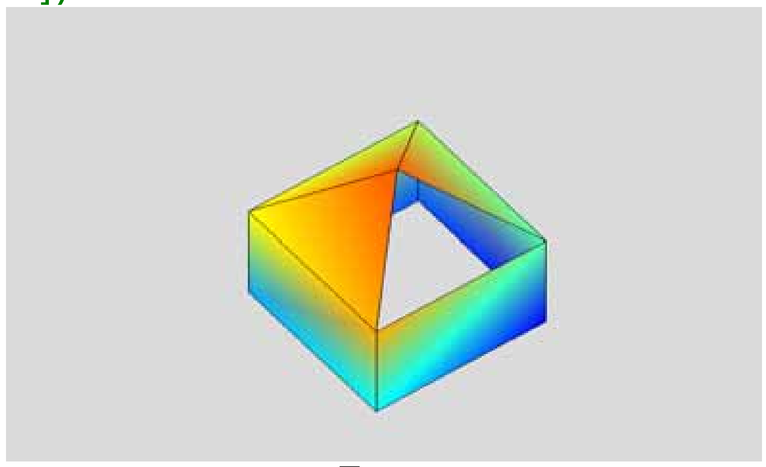


图 9.7-12

9.7.9 鼠标拖动字对象

【例 9.7.9-1】本程序有两个功能：把当前图形窗中已存在的任何字对象拖放到所需的位置；向当前图形窗输入字对象，并拖放到任何所需位置。

（1）

[textzzy.m]

```
function textzzy(arg)
% textzzy.m
%
%
if ~nargin;arg=0;end
if ischar(arg)|iscell(arg)
   PT.Units='normalized';
   PT.Position=[0.01 0.01];
   PT.String=arg;
   PT.HorizontalAlignment='left';
   PT.VerticalAlignment='baseline';
   ht=text(PT);
   textzzy(0)
elseif arg==0
   hf=get(0,'CurrentFigure');
   if isempty(hf)
      error('图形窗不存在。')
   end
   PF1.BackingStore='off';
   PF1.WindowButtonDownFcn='textzzy(1)';
   set(hf,PF1)
   figure(hf)
elseif arg==1 & strcmp(get(gco,'Type'),'text')
   PO1.Units='data';
   PO1.EraseMode='xor';
   PO1.HorizontalAlignment='left';PO1.VerticalAlignment='baseline';
   set(gco,PO1)
   PF2.Pointer='fleur';
   PF2.WindowButtonMotionFcn='textzzy(2)';
   PF2.WindowButtonUpFcn='textzzy(999)';
   set(gcf,PF2)
elseif arg==2
   curpoi=get(gca,'CurrentPoint');
   set(gco,'Position',curpoi(1,1:3))
elseif arg==999
   set(gco,'EraseMode','normal')
   PF3.WindowButtonDownFcn='';
   PF3.WindowButtonMotionFcn='';
   PF3.WindowButtonUpFcn='';
   PF3.Pointer='arrow';
   PF3.Units='pixels';
   PF3.BackingStore='on';
   set(gcf,PF3)
else
   PF4.WindowButtonDownFcn='';
   PF4.WindowButtonMotionFcn='';
   PF4.WindowButtonUpFcn='';
   PF4.Pointer='arrow';
   PF4.Units='pixels';
   PF4.BackingStore='on';
   set(gcf,PF4)
```

```
end
```

(2)

(3)

第十章 图形用户界面 GUI 制作

用户界面（或接口）是指：人与机器（或程序）之间交互作用的工具和方法。如键盘、鼠标、跟踪球、话筒都可成为与计算机交换信息的接口。

图形用户界面（Graphical User Interfaces ，GUI）则是由窗口、光标、按键、菜单、文字说明等对象（Objects）构成的一个用户界面。用户通过一定的方法（如鼠标或键盘）选择、激活这些图形对象，使计算机产生某种动作或变化，比如实现计算、绘图等。

假如读者所从事的数据分析、解方程、计算结果可视工作比较单一，那么一般不会考虑 GUI 的制作。但是如果读者想向别人提供应用程序，想进行某种技术、方法的演示，想制作一个供反复使用且操作简单的专用工具，那么图形用户界面也许是最好的选择之一。

MATLAB 为表现其基本功能而设计的演示程序 demo 是使用图形界面的最好范例。MATLAB 的用户，在指令窗中运行 demo 打开那图形界面后，只要用鼠标进行选择和点击，就可浏览那丰富多彩的内容。

即便比较熟悉 MATLAB 的读者，在他初次编写 GUI 程序时，也会感到棘手。为使读者获得制作自己 GUI 的体验，本章“入门”节提供了一个简单的示例。读者只要输入所提供的程序，就可引出相应的界面。

本章第 2 节叙述图形用户界面的设计原则和一般制作步骤。第 3、4 节分别介绍用户菜单、用户控件的制作。出于“由浅入深”的考虑，前 4 节制作 GUI 是通过 M 脚本文件实现的。利用 M 函数文件制作 GUI，需要解决数据传递问题，为此专设第 5 节给予阐述和示例。这前 5 节内容对读者理解交互图形界面的工作原理很有帮助。

本章第 6 节专述 MATLAB6.5 提供的界面设计工作台的使用。值得指出：该设计工作台与 MATLAB5.3 版的设计工具有很大不同。新的设计工作台显得更成熟、方便。

在此提醒读者，假如要比较准确的理解本章程序和掌握本章内容，请先阅读第 9 章关于图柄的内容。

10.1入门

【例 10.1-1】对于传递函数为 $G=\dfrac{1}{s^2+2\zeta s+1}$ 的归一化二阶系统，制作一个能绘制该系统单位阶跃响应的图形用户界面。本例演示：（A）图形界面的大致生成过程；（B）静态文本和编辑框的生成；（C）坐标方格控制键的形成；（D）如何使用该界面。

（1）

```
clf reset
H=axes('unit','normalized','position',[0,0,1,1],'visible','off');
set(gcf,'currentaxes',H);
str='\fontname{隶书}归一化二阶系统的阶跃响应曲线';
text(0.12,0.93,str,'fontsize',13);
h_fig=get(H,'parent');
set(h_fig,'unit','normalized','position',[0.1,0.2,0.7,0.4]);
h_axes=axes('parent',h_fig,...
   'unit','normalized','position',[0.1,0.15,0.55,0.7],...
   'xlim',[0 15],'ylim',[0 1.8],'fontsize',8);
```

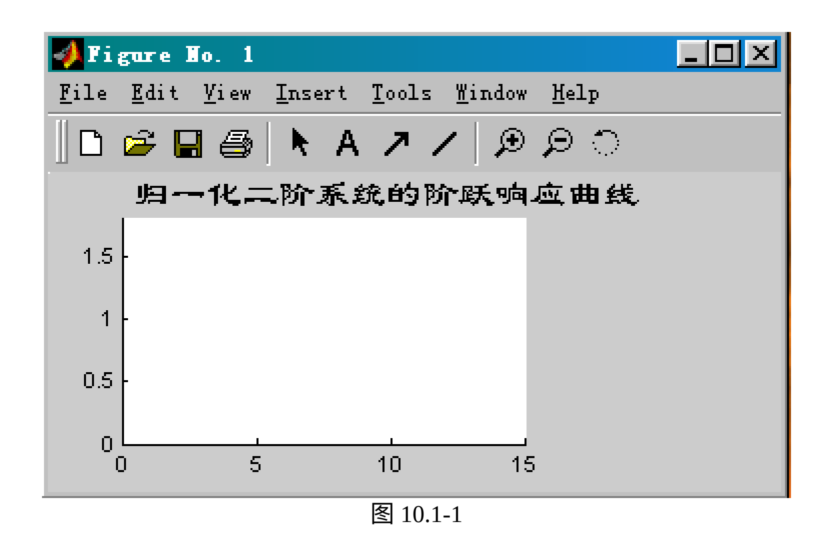


图 10.1-1

(2)

```
h_text=uicontrol(h_fig,'style','text',...
   'unit','normalized','position',[0.67,0.73,0.25,0.14],...
   'horizontal','left','string',{'输入阻尼比系数','zeta ='});
h_edit=uicontrol(h_fig,'style','edit',...
   'unit','normalized','position',[0.67,0.59,0.25,0.14],...
   'horizontal','left',...
   'callback',[...
      'z=str2num(get(gcbo,''string''));',...
      't=0:0.1:15;',...
      'for k=1:length(z);',...
      'y(:,k)=step(1,[1 2*z(k) 1],t);',...
      'plot(t,y(:,k));',...
      'if (length(z)>1) ,hold on,end,',...
      'end;',...
    'hold off,']);
```

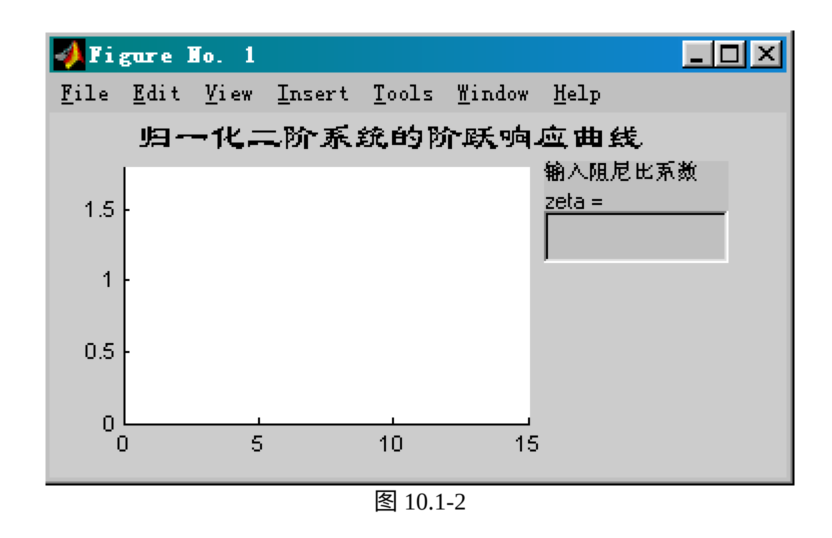


图 10.1-2

(3)

```
h_push1=uicontrol(h_fig,'style','push',...
   'unit','normalized','position',[0.67,0.37,0.12,0.15],...
   'string','grid on','callback','grid on');
h_push2=uicontrol(h_fig,'style','push',...
```

```
'unit','normalized','position',[0.67,0.15,0.12,0.15],...
'string','grid off','callback','grid off');
```

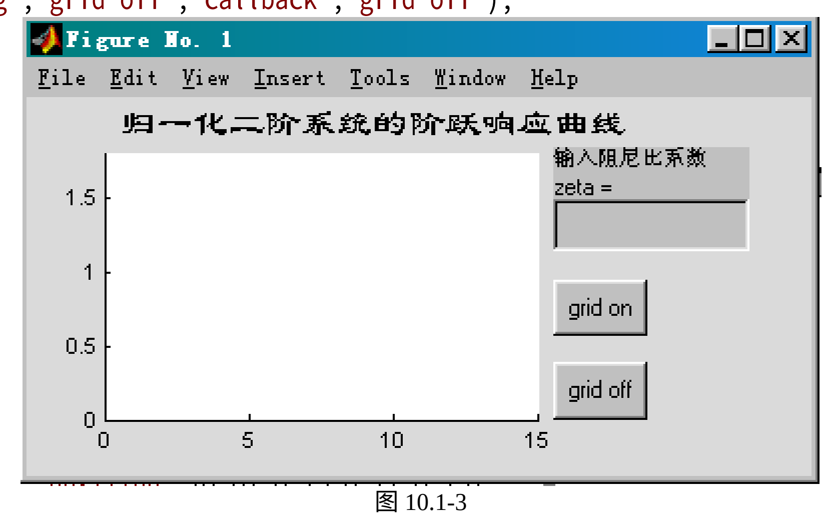


图 10.1-3

(4)

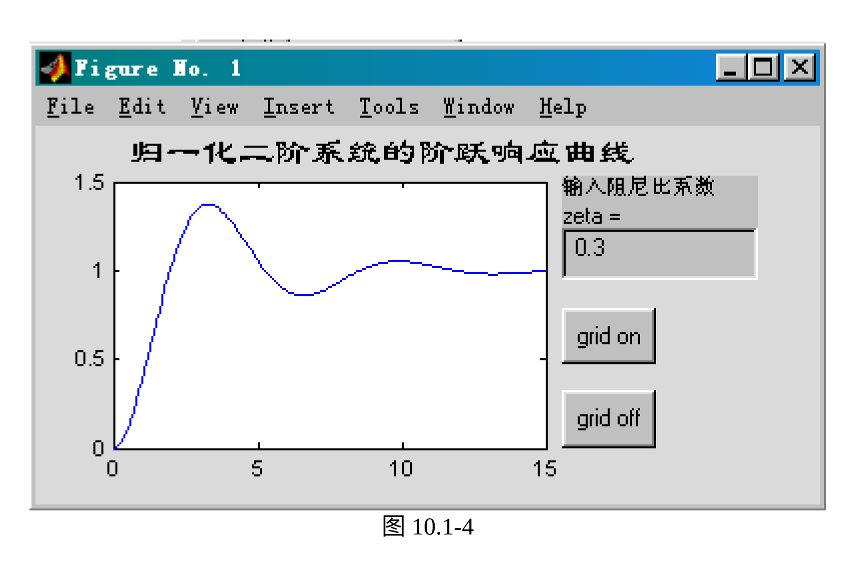


图 10.1-4

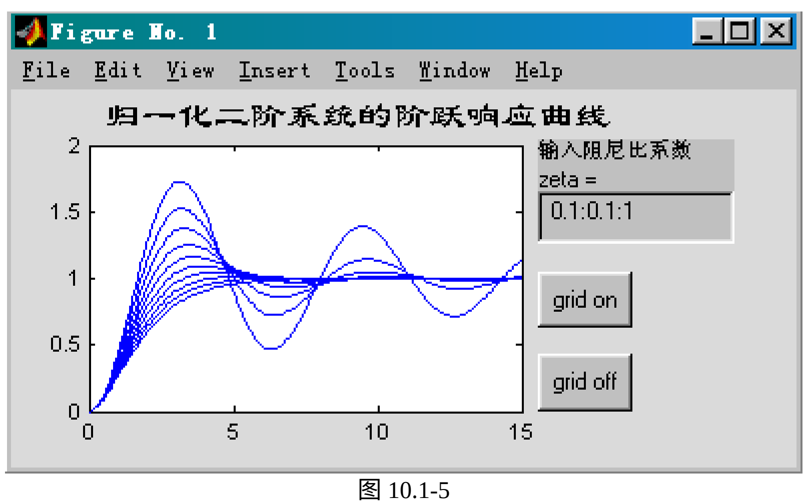


图 10.1-5

10.2图形用户界面的设计原则和一般步骤

10.2.1设计原则

（1）简单性
（2）一致性
（3）习常性
（4）其它考虑因素

10.2.2 一般制作步骤

10.3界面菜单（uimenu）

10.3.1图形窗的标准菜单

【例 10.3.1-1】本例说明：如何隐藏和恢复标准菜单的显示。

（1）

H_fig=figure

（2）

set(H_fig , 'MenuBar','none')；
set(gcf,'menubar',menubar);

（3）

set(gcf,'menubar','figure');

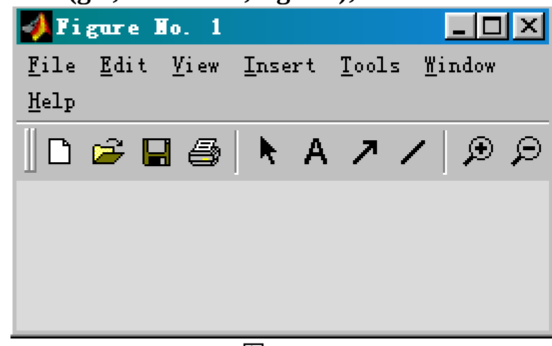


图 10.3-1

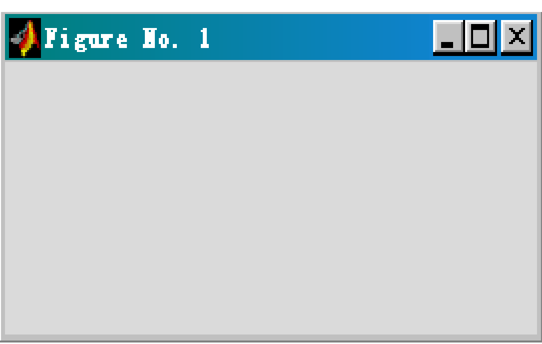


图 10.3-2

10.3.2自制的用户菜单

【例 10.3.2-1】本例演示：如何自制一个带下拉菜单表的用户菜单（如图 10.3-3 所示）。该菜单能使图形窗背景颜色设置为兰色或红色。

```
figure
h_menu=uimenu(gcf,'label','Color');                    %    <2>
h_submenu1=uimenu(h_menu,'label','Blue',...            %    <3>
    'callback','set(gcf,''Color'',''blue'')');         %    <4>
h_submenu2=uimenu(h_menu,'label','Red',...             %    <5>
    'callback','set(gcf,''Color'',''red'')');          %     <6>
```

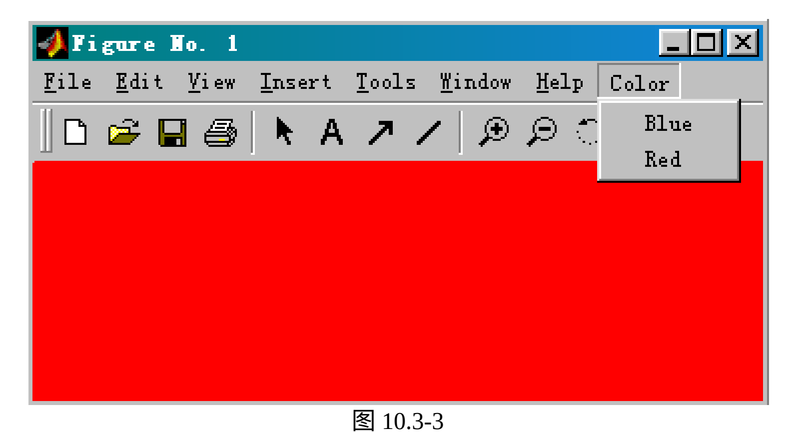


图 10.3-3

10.3.3 用户菜单的属性

10.3.3.1 回调属性和菜单名

（1）菜单名属性

（2）回调属性

【例 10.3.3.1-1】本例的目标是：在图形窗上自制一个名为【Test】的“顶层菜单项”；当用鼠标点动该菜单项时，将产生一个带分格的封闭坐标轴。通过本例说明：（A）回调属性的运作机理；（B）用户顶层菜单项的制作（C）uimenu 属性的设置方法；（D）复杂字符串的构成方法和注意事项。

（1）

```
grid on,set(gca,'box','on')
```

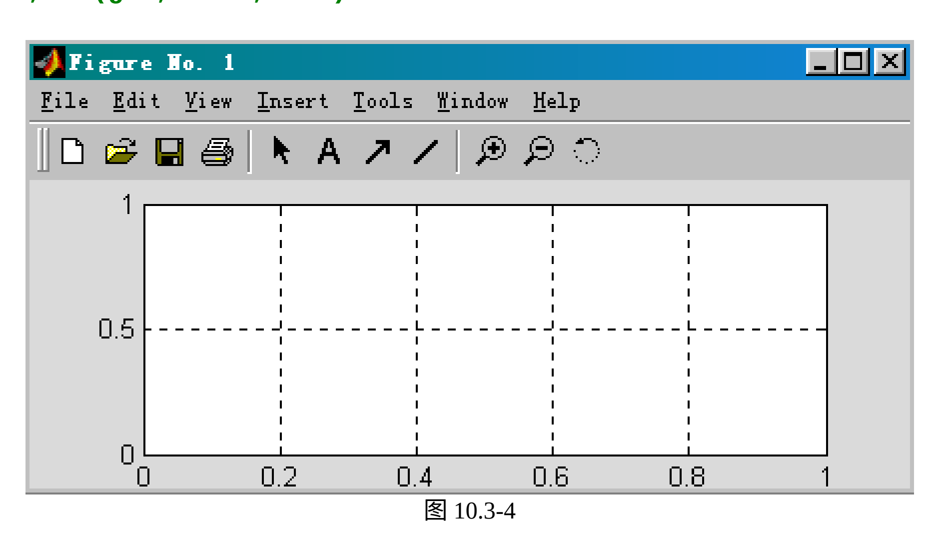


图 10.3-4

（2）

```
eval('grid on,set(gca,''box'',''on'')')
```

（3）

```
uimenu('Label','Test','Callback','grid on,set(gca,''box'',''on''),')
```

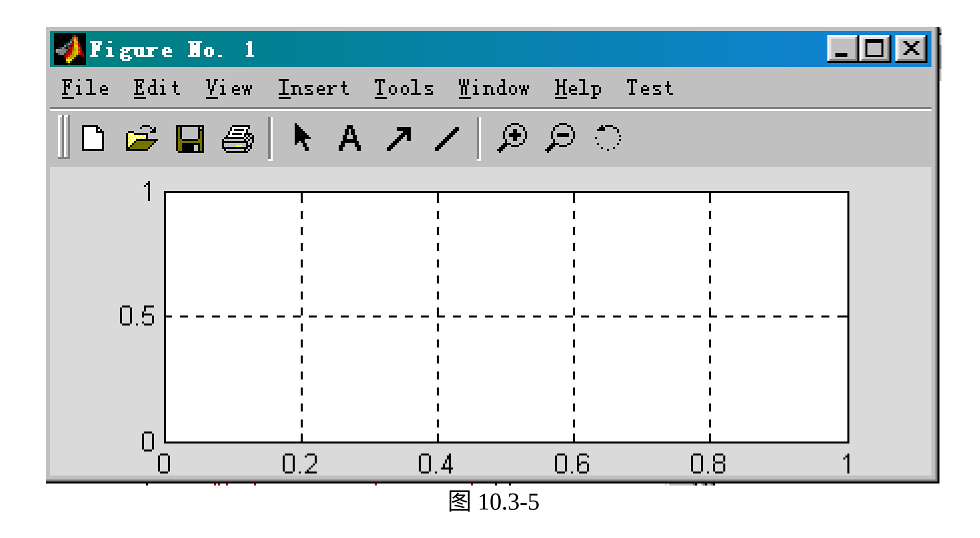


图 10.3-5

(4)
```
uimenu('Label','Test', ...
       'Callback',['grid on,' , ...
                   'set(gca,''box'',''on'');'])
```

(5)
```
Lpv='Test';
Cpv=['grid on,','set(gca,''box'',''on''),'];
uimenu('Label', Lpv, 'Callback' , Cpv)
```

(6)
```
PS.Label='Test';
PS.Callback=['grid on;','set(gca,''box'',''on'');'];
uimenu(PS)
```

10.3.3.2　设置简捷键或快捷键

【例 10.3.3.2-1】本例目标：使图 10.3-3 所示菜单成为图 10.3-6 那样，Color 菜单项及其下拉的 Blue 菜单各带一个简捷键，而另一项下拉菜单 Red 带一个快捷键。

[exm100332_1.m]
```
figure
h_menu=uimenu(gcf,'Label','&Color');           %   <2>
h_submenu1=uimenu(h_menu,'Label','&Blue',... %   <3>
    'Callback','set(gcf,''color'',''blue'')');
h_submenu2=uimenu(h_menu,'label','Red',...
    'Callback','set(gcf,''color'',''red'')',...
    'Accelerator','r');                         %   <7>
```

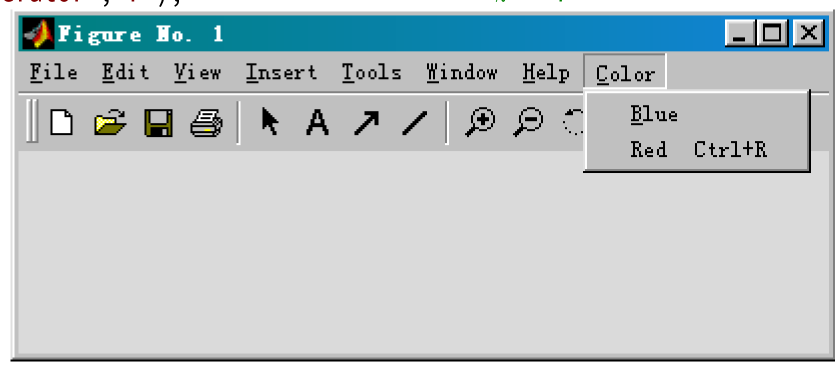


图 10.3-6

10.3.3.3　用户菜单的外观设计

【例 10.3.3.3-1】本例演示：（A）把用户菜单 'Option' 设置为顶层的第 3 菜单项；（B）下拉菜单被两条分隔线分为三个菜单区；（C）最下菜单项又有两个子菜单组成。

（1）

[exm100333_1.m]

```
figure
BackColor=get(gcf,'Color');
h_menu=uimenu('label','Option','Position',3);
h_sub1=uimenu(h_menu,'label','grid on','callback','grid on');
h_sub2=uimenu(h_menu,'label','grid off','callback','grid on');
h_sub3=uimenu(h_menu,'label','box on','callback','box on',...
    'separator','on');                                              %<6>
h_sub4=uimenu(h_menu,'label','box off','callback','box off');
h_sub5=uimenu(h_menu,'label','Figure Color','Separator','on');      %<8>
h_subsub1=uimenu(h_sub5,'label','Red','ForeGroundColor','r',...     %<9>
   'callback','set(gcf,''Color'',''r'')');
h_subsub2=uimenu(h_sub5,'label','Reset',...
   'callback','set(gcf,''Color'',BackColor)');
```

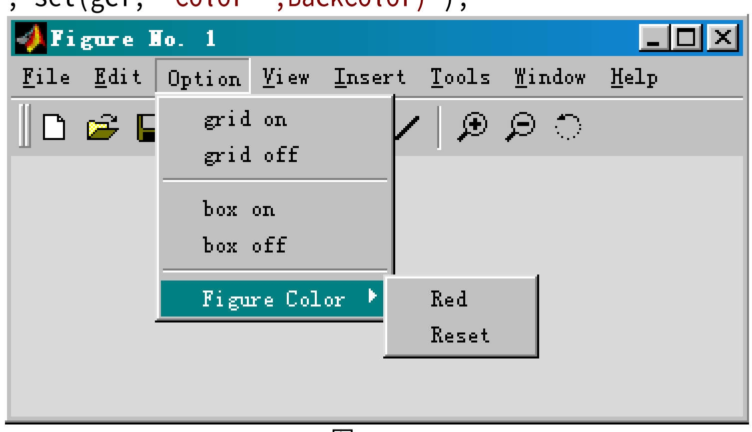


图 10.3-7

（2）

```
Pos_O=get(h_menu,'position'),             %查询 Option 菜单位置值
Pos_BoxOn=get(h_sub3,'position')          %查询 box ob 子菜单位置值
Pos_Red=get(h_subsub1,'position')       %查询 red 子菜单的位置值
Pos_O =
     3
Pos_BoxOn =
     3
Pos_Red =
     1
```

【例 10.3.3.3-2】本例演示：当某菜单项选中后，如何使该菜单项贴上检录符 “√”。

[exm100333_2.m]

```
figure
h_menu=uimenu('label','Option');
h_sub1=uimenu(h_menu,'label','Grid on',...                        %<3>
```

```
    'callback',[...
     'grid on,',...
     'set(h_sub1,''checked'',''on''),',...
     'set(h_sub2,''checked'',''off''),',...
    ]);
h_sub2=uimenu(h_menu,'label','Grid off',...                                    %<4>
    'callback',[...
     'grid off,',...
     'set(h_sub2,''checked'',''on''),',...
     'set(h_sub1,''checked'',''off''),',...
     ]);
```

图 10.3-8

10.3.3.4　使能（Enable）与可见性（Visible）属性

【例 10.3.3.4-1】 本例目标：制作一个带四个子菜单项的顶层菜单项；该下拉菜单分为两个功能区；每个功能区的两个菜单项是相互对立的，因此采用使能属性处理；当图形窗坐标轴消隐时，整个坐标分隔控制功能区不可见。

（1）

[exm100334_1.m]

```
clf
h_menu=uimenu('label','Option');
h_sub1=uimenu(h_menu,'label','Axis on');
h_sub2=uimenu(h_menu,'label','Axis off',...
   'enable','off');
h_sub3=uimenu(h_menu,'label','Grid on',...
   'separator','on','visible','off');
h_sub4=uimenu(h_menu,'label','Grid off',...
   'visible','off');
set(h_sub1,'callback',[...
   'Axis on,',...
   'set(h_sub1,''enable'',''off''),',...
   'set(h_sub2,''enable'',''on''),',...
   'set(h_sub3,''visible'',''on''),',...
   'set(h_sub4,''visible'',''on''),']);
set(h_sub2,'callback',[...
   'axis off,',...
   'set(h_sub1,''enable'',''on''),',...
   'set(h_sub2,''enable'',''off''),',...
   'set(h_sub3,''visible'',''off''),',...
```

```
    'set(h_sub4,''visible'',''off''),']);
set(h_sub3,'callback',[...
    'grid on,',...
    'set(h_sub3,''enable'',''off''),',...
    'set(h_sub4,''enable'',''on''),']);
set(h_sub4,'callback',[...
    'grid off,',...
    'set(h_sub3,''enable'',''on''),',...
    'set(h_sub4,''enable'',''off''),']);
```

(2)

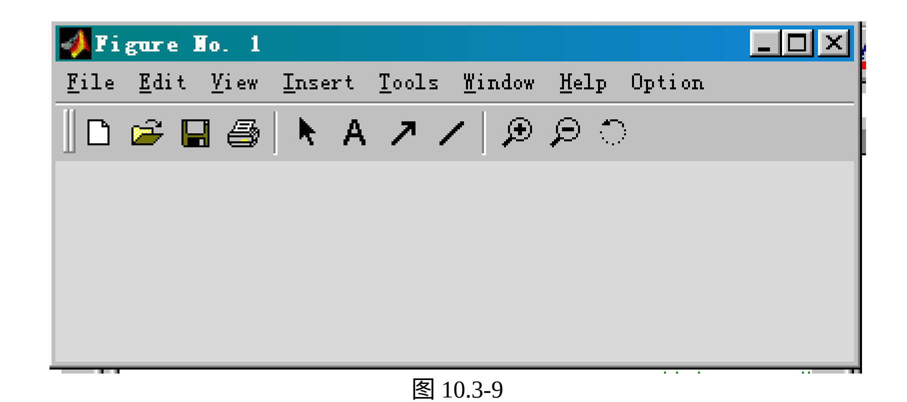


图 10.3-9

(3)

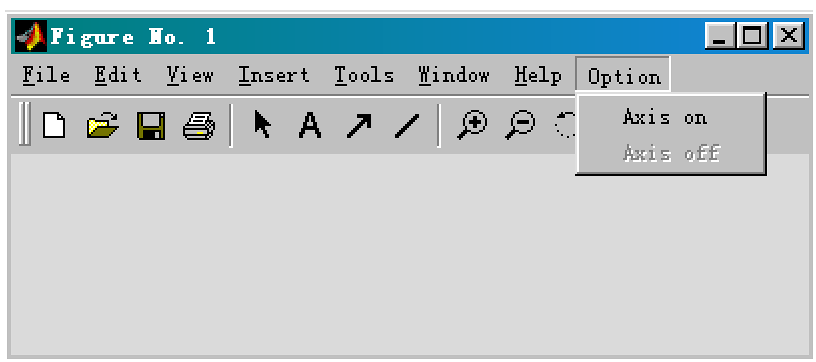


图 10.3-10

(4)

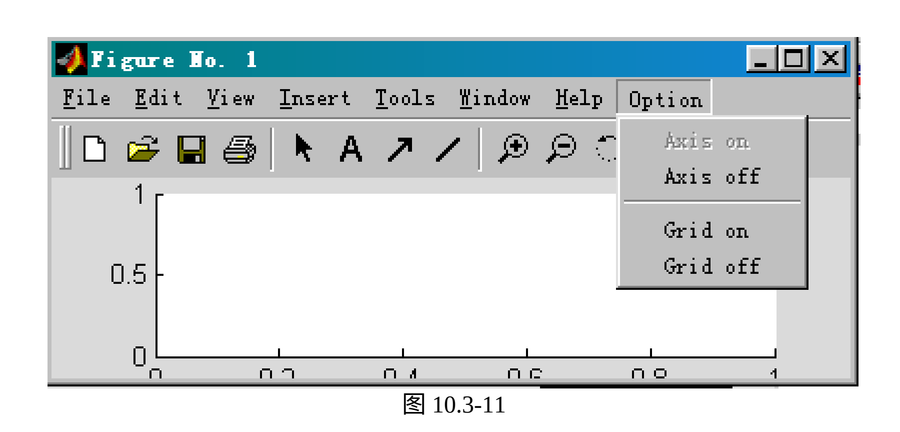


图 10.3-11

(5)

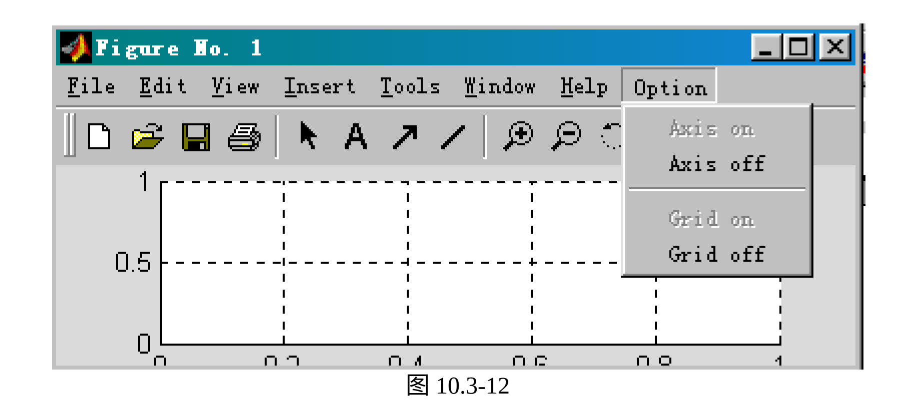


图 10.3-12

10.3.4 现场菜单的制作

【例 10.3.4-1】目标：绘制一条 Sa 曲线，创建一个与之相联系的现场菜单，用以控制 Sa 曲线的颜色。

（1）

[exm10034_1.m]

```
t=(-3*pi:pi/50:3*pi)+eps;
y=sin(t)./t;
hline=plot(t,y);
cm=uicontextmenu;
%
uimenu(cm,'label','Red','callback','set(hline,''color'',''r''),')
uimenu(cm,'label','Blue','callback','set(hline,''color'',''b''),')
uimenu(cm,'label','Green','callback','set(hline,''color'',''g''),')
set(hline,'uicontextmenu',cm)
```

（2）

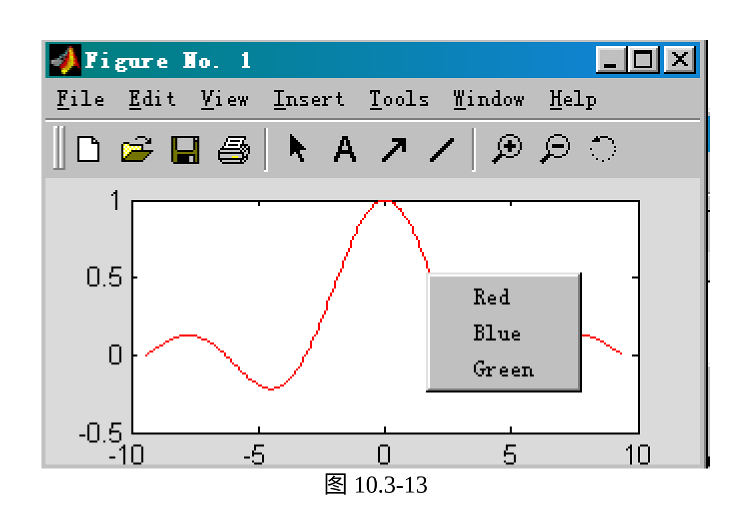


图 10.3-13

10.4用户控件（uicontrol）

10.4.1 控件制作函数

10.4.2 用户控件的种类

10.4.3 控件制作示例

10.4.3.1　双位按键、无线电按键、控件区域框示例

【例 10.4.3.1-1】目标：创建一个界面包含 4 种控件：静态文本、“无线电”选择开关、双位按键、控件区域框。

[exm100431_1.m]

```
clf reset
set(gcf,'menubar','none')
set(gcf,'unit','normalized','position',[0.2,0.2,0.64,0.32]);
set(gcf,'defaultuicontrolunits','normalized')
h_axes=axes('position',[0.05,0.2,0.6,0.6]);
t=0:pi/50:2*pi;y=sin(t);plot(t,y);
set(h_axes,'xlim',[0,2*pi]);
set(gcf,'defaultuicontrolhorizontal','left');
htitle=title('正弦曲线');
set(gcf,'defaultuicontrolfontsize',12);
uicontrol('style','frame',...                          %  <11>
   'position',[0.67,0.55,0.25,0.25]);
uicontrol('style','text',...                           %  <13>
   'string','正斜体图名:',...
   'position',[0.68,0.77,0.18,0.1],...
   'horizontal','left');
hr1=uicontrol(gcf,'style','radio',...                  %  <17>
   'string','正体',...
   'position',[0.7,0.69,0.15,0.08]);
set(hr1,'value',get(hr1,'Max'));                       %  <20>
set(hr1,'callback',[...                                %  <21>
       'set(hr1,''value'',get(hr1,''max'')),',...      %  <22>
       'set(hr2,''value'',get(hr2,''min'')),',...      %  <23>
       'set(htitle,''fontangle'',''normal''),',...
    ]);
hr2=uicontrol(gcf,'style','radio',...                  %  <26>
    'string','斜体',...
    'position',[0.7,0.58,0.15,0.08],...
    'callback',[...
       'set(hr1,''value'',get(hr1,''min'')),',...      %  <30>
       'set(hr2,''value'',get(hr2,''max'')),',...      %  <31>
       'set(htitle,''fontangle'',''italic'')',...
    ]);                                                %  <33>
   ht=uicontrol(gcf,'style','toggle',...               %  <34>
      'string','Grid',...
      'position',[0.67,0.40,0.15,0.12],...
      'callback','grid');
```

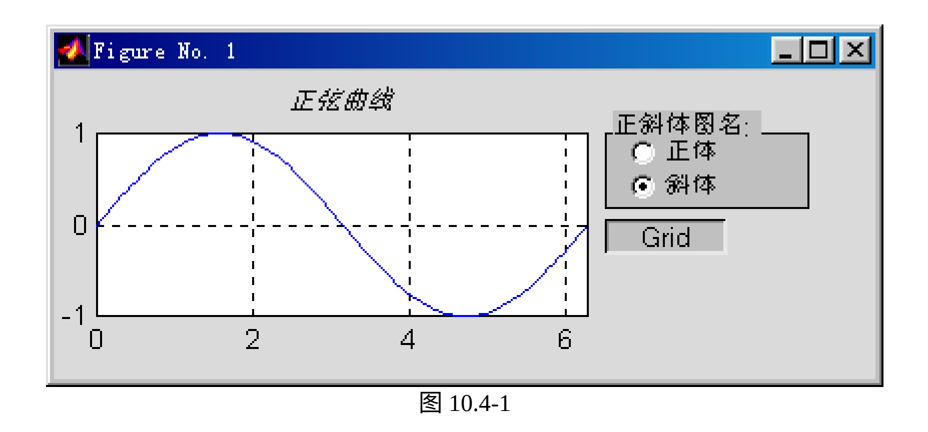


图 10.4-1

10.4.3.2 静态文本框、滑动键、检录框示例

【例 10.4.3.2-1】目标：制作演示“归一化二阶系统单位阶跃响应”的交互界面。在该界面中，阻尼比可在[0.02,2.02]中连续调节，标志当前阻尼比值；可标志峰值时间和大小；可标志（响应从 0 到 0.95 所需的）上升时间。本例涉及以下主要内容：（A）静态文本的创建和实时改写。（B）滑动键的创建；'Max' 和 'Min' 的设置；'Value' 的设置和获取。（C）检录框的创建；'Value' 的获取。（D）受多个控件影响的回调操作。

[exm100432_1.m]

```
clf reset
set(gcf,'unit','normalized','position',[0.1,0.2,0.64,0.35]);
set(gcf,'defaultuicontrolunits','normalized');
set(gcf,'defaultuicontrolfontsize',12);
set(gcf,'defaultuicontrolfontname','隶书');
set(gcf,'defaultuicontrolhorizontal','left');
str='归一化二阶系统阶跃响应曲线';
set(gcf,'name',str,'numbertitle','off');
h_axes=axes('position',[0.05,0.2,0.6,0.7]);
set(h_axes,'xlim',[0,15]);
str1='当前阻尼比=';
t=0:0.1:10;z=0.5;y=step(1,[1 2*z 1],t);
hline=plot(t,y);
htext=uicontrol(gcf,'style','text',...        %  <14>
   'position',[0.67,0.8,0.33,0.1],...
   'string',[str1,sprintf('%1.4g\',z)]);
hslider=uicontrol(gcf,'style','slider',...    %  <17>
   'position',[0.67,0.65,0.33,0.1],...
   'max',2.02,'min',0.02,...                  %  <19>
   'sliderstep',[0.01,0.05],...               %  <20>
   'Value',0.5);                              %  <21>
hcheck1=uicontrol(gcf,'style','checkbox',...%  <22>
   'string','最大峰值' ,...
   'position',[0.67,0.50,0.33,0.11]);
vchk1=get(hcheck1,'value');                   %  <25>
hcheck2=uicontrol(gcf,'style','checkbox',...%  <26>
   'string','上升时间(0->0.95)',...
   'position',[0.67,0.35,0.33,0.11]);
vchk2=get(hcheck2,'value');                   %  <29>
set(hslider,'callback',[...                   %  <30>
```

```
    'z=get(gcbo,''value'');',...                    %  <31>
    'callcheck(htext,str1,z,vchk1,vchk2)']);  %  <32>
set(hcheck1,'callback',[...                        %  <33>
    'vchk1=get(gcbo,''value'');',...               %  <34>
    'callcheck(htext,str1,z,vchk1,vchk2)']);  %  <35>
set(hcheck2,'callback',[...                        %  <36>
    'vchk2=get(gcbo,''value'');',...               %  <37>
    'callcheck(htext,str1,z,vchk1,vchk2)']);  %  <38>
```

[callcheck.m]

```
function callcheck(htext,str1,z,vchk1,vchk2)
cla,set(htext,'string',[str1,sprintf('%1.4g\',z)]);    %  <2>
dt=0.1;t=0:dt:15;N=length(t);y=step(1,[1 2*z 1],t);plot(t,y);
if vchk1                                                %  <4>
   [ym,km]=max(y);
   if km<(N-3)                                          %  <6>
      k1=km-3;k2=km+3;k12=k1:k2;tt=t(k12);
      yy=spline(t(k12),y(k12),tt);                      %  <8>
      [yym,kkm]=max(yy);
      line(tt(kkm),yym,'marker','.',...                 %  <10>
         'markeredgecolor','r','markersize',20);
      ystr=['ymax = ',sprintf('%1.4g\',yym)];
      tstr=['tmax = ',sprintf('%1.4g\',tt(kkm))];
      text(tt(kkm),1.05*yym,{ystr;tstr})
   else                                                 %  <15>
      text(10,0.4*y(end),{'ymax --> 1';'tmax --> inf'})
   end
end
if vchk2                                                %  <19>
   k95=min(find(y>0.95));k952=[(k95-1),k95];
   t95=interp1(y(k952),t(k952),0.95);                   %  <21>
   line(t95,0.95,'marker','o','markeredgecolor','k','markersize',6);
   tstr95=['t95 = ',sprintf('%1.4g\',t95)];
   text(t95,0.65,tstr95)
end
```

图 10.4-2

10.4.3.3　可编辑框、弹出框、列表框、按键示例

【例 10.4.3.3-1】目标：制作一个能绘制任意图形的交互界面。它包括：可编辑文本框、弹出框、列表框。本例的关键内容是：如何使编辑框允许输入多行指令。

[exm100433_1.m]

```
clf reset                                              %                    <1>
set(gcf,'unit','normalized','position',[0.1,0.4,0.85,0.35]);
set(gcf,'defaultuicontrolunits','normalized');
set(gcf,'defaultuicontrolfontsize',11);
set(gcf,'defaultuicontrolfontname','隶书');
set(gcf,'defaultuicontrolhorizontal','left');
set(gcf,'menubar','none');
str='通过多行指令绘图的交互界面';
set(gcf,'name',str,'numbertitle','off');
h_axes=axes('position',[0.05,0.15,0.45,0.70],'visible','off');
uicontrol(gcf,'Style','text',...
   'position',[0.52,0.87,0.26,0.1],...
   'String','绘图指令输入框');
hedit=uicontrol(gcf,'Style','edit',...             %   <14>
   'position',[0.52,0.05,0.26,0.8],...
   'Max',2);                                 %   <16>
hpop=uicontrol(gcf,'style','popup',...             %   <17>
   'position',[0.8,0.73,0.18,0.12],...
   'string','spring|summer|autumn|winter');%<19>
hlist=uicontrol(gcf,'Style','list',...             %   <20>
   'position',[0.8,0.23,0.18,0.37],...
   'string','Grid on|Box on|Hidden off|Axis off',...%  <22>
   'Max',2);                                       %   <23>
hpush=uicontrol(gcf,'Style','push',...             %   <24>
   'position',[0.8,0.05,0.18,0.15],'string','Apply');
set(hedit,'callback','calledit(hedit,hpop,hlist)');%   <26>
set(hpop,'callback','calledit(hedit,hpop,hlist)'); %   <27>
set(hpush,'callback','calledit(hedit,hpop,hlist)');%   <28>
```

[calledit.m]

```
function calledit(hedit,hpop,hlist)
ct=get(hedit,'string');               %   <2>
vpop=get(hpop,'value');               %   <3>
vlist=get(hlist,'value');             %   <4>
if ~isempty(ct)                       %   <5>
   eval(ct')                          %   <6>
   popstr={'spring','summer','autumn','winter'};      %   <7>
   liststr={'grid on','box on','hidden off','axis off'};% <8>
   invstr={'grid off','box off','hidden on','axis on'};%  <9>
   colormap(eval(popstr{vpop}))       %   <10>
   vv=zeros(1,4);vv(vlist)=1;
   for k=1:4
      if vv(k);eval(liststr{k});else eval(invstr{k});end
   end
end
```

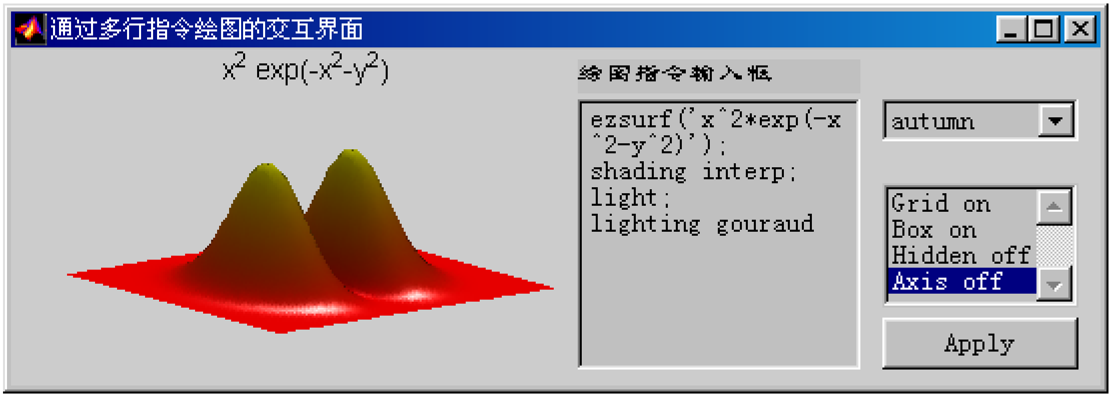


图 10.4-3

10.5由 M 函数文件产生用户菜单和控件

10.5.1利用全局变量编写用户界面函数文件

【例 10.5.1-1】目标：利用 M 函数文件创建与例 10.4.3.3-1 相同的用户界面。本例演示：如何依靠全局变量传递控件的图柄，从而保证回调动作正确执行。

（1）

[exm10051_1.m]

```
function exm10051_1( )
global hedit hpop hlist
```

（这中间是：原 exm100433_1.m 第〈1〉行到第〈25〉行的全部指令）

```
set(hedit,'callback','calledit1');                % <26>
set(hpop,'callback','calledit1');                 % <27>
set(hpush,'callback','calledit1');                % <28>
```

[calledit1.m]

```
function calledit1( )
global hedit hpop hlist
```

（下面续接内容是：原 calledit.m 第〈2〉行以下的全部指令）

（2）

10.5.2利用 'UserData' 属性编写用户界面函数文件

【例 10.5.2-1】目标：利用 M 函数文件创建与例 10.4.3.3-1 相同的用户界面。本例演示：如何依靠图形窗的'UserData' 属性传送用户控件的图柄，从而保证回调动作正确执行。

（1）

[exm10052_1.m]

```
function exm10052_1( )
```

（这中间是：原 exm100433_1.m 第〈1〉行到第〈25〉行的全部指令）

```
set(hedit,'callback','calledit2');                % <26>
set(hpop,'callback','calledit2');                 % <27>
set(hpush,'callback','calledit2');                % <28>
set(gcf,'UserData',[hedit,hpop,hlist])
```

[calledit2.m]

```
function calledit2( )
H=get(gcf,'UserData');
ct=get(H(1),'string');              %   <2>
vpop=get(H(2),'value');             %   <3>
vlist=get(H(3),'value');            %   <4>
```

（下面续接内容是：原 calledit.m 第〈5〉行以下的全部指令）

（2）

10.5.3利用递归法编写用户界面函数文件

【例 10.5.3-1】目标：利用 M 函数文件创建与例 10.4.3.3-1 相同的用户界面。本例演示：如何依靠图形窗'UserData' 属性在递归调用中传送用户控件的图柄，保证回调动作正确执行。

（1）

[exm10053_1.m]

```
function exm10053_1(flag)
if nargin<1;flag='startup';end                              %   <2>
if ~ischar(flag);error('flag must be character ''startup''.');end
switch flag                                                 %   <4>
case 'startup'                                              %   <5>
clf reset                                                   %   <6>
set(gcf,'unit','normalized','position',[0.1,0.4,0.85,0.35]);
set(gcf,'defaultuicontrolunits','normalized');
set(gcf,'defaultuicontrolfontsize',11);
set(gcf,'defaultuicontrolfontname','隶书');
set(gcf,'defaultuicontrolhorizontal','left');
set(gcf,'menubar','none');
str='通过多行指令绘图的交互界面';
set(gcf,'name',str,'numbertitle','off');
h_axes=axes('position',[0.05,0.15,0.45,0.70],'visible','off');
uicontrol(gcf,'Style','text',...
   'position',[0.52,0.87,0.26,0.1],...
   'String','绘图指令输入框');
hedit=uicontrol(gcf,'Style','edit',...                      %   <19>
   'position',[0.52,0.05,0.26,0.8],...                      %   <20>
   'Max',2);                                                %   <21>
hpop=uicontrol(gcf,'style','popup',...                      %   <22>
   'position',[0.8,0.73,0.18,0.12],...                      %   <23>
   'string','spring|summer|autumn|winter');                 %   <24>
hlist=uicontrol(gcf,'Style','list',...                      %   <25>
   'position',[0.8,0.23,0.18,0.37],...                      %   <26>
   'string','Grid on|Box on|Hidden off|Axis off',...        %   <27>
   'Max',2);                                                %   <28>
hpush=uicontrol(gcf,'Style','push',...                      %   <29>
   'position',[0.8,0.05,0.18,0.15],'string','Apply');
set(hedit,'callback','exm1053_1(''set'')');                 %   <31>
set(hpop,'callback','exm1053_1(''set'')');                  %   <32>
```

```
set(hpush,'callback','exm1053_1(''set'')');                    % <33>
set(gcf,'UserData',[hedit,hpop,hlist]);                        % <34>
case 'set'                                                     % <35>
H=get(gcf,'UserData');                                         % <36>
ct=get(H(1),'string');                                         % <37>
vpop=get(H(2),'value');                                        % <38>
vlist=get(H(3),'value');                                       % <39>
if ~isempty(ct)
   eval(ct')
   popstr={'spring','summer','autumn','winter'};
   liststr={'grid on','box on','hidden off','axis off'};
   invstr={'grid off','box off','hidden on','axis on'};
   colormap(eval(popstr{vpop}))
   vv=zeros(1,4);vv(vlist)=1;
   for k=1:4
      if vv(k);eval(liststr{k});else eval(invstr{k});end
   end
end                                                            % <50>
end
```

(2)

【例 10.5.3-2】目标：利用 M 函数文件创建与例 10.4.3.3-1 相同的用户界面。本例演示：如何依靠 'Tag'属性 与 findobj 指令的配合使用获取回调操作所必须的控件图柄，保证回调动作正确执行。

(1)

(2)

```
   'Tag','H_edit',...
```

(3)

```
   'Tag','H_popup',...
```

(4)

```
   'Tag','H_list',...
```

(5)

(6)

```
H(1)=findobj(gcf,'Tag','H_edit');
H(2)=findobj(gcf,'Tag','H_popup');
H(3)=findobj(gcf,'Tag','H_list');
```

(7)

```
function exm10053_2(flag)
```

(8)

(9)

10.6图形用户界面设计工具

10.6.1 界面设计工具的结构和调用指令

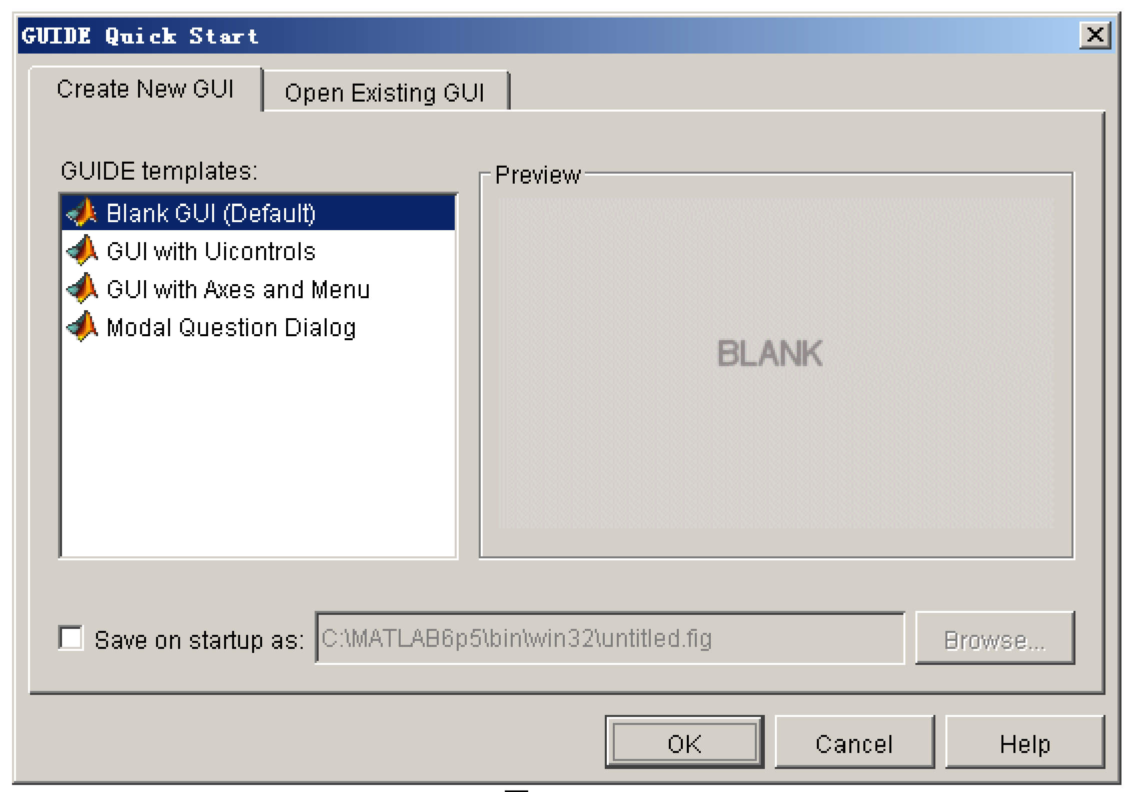


图 10.6.1-1

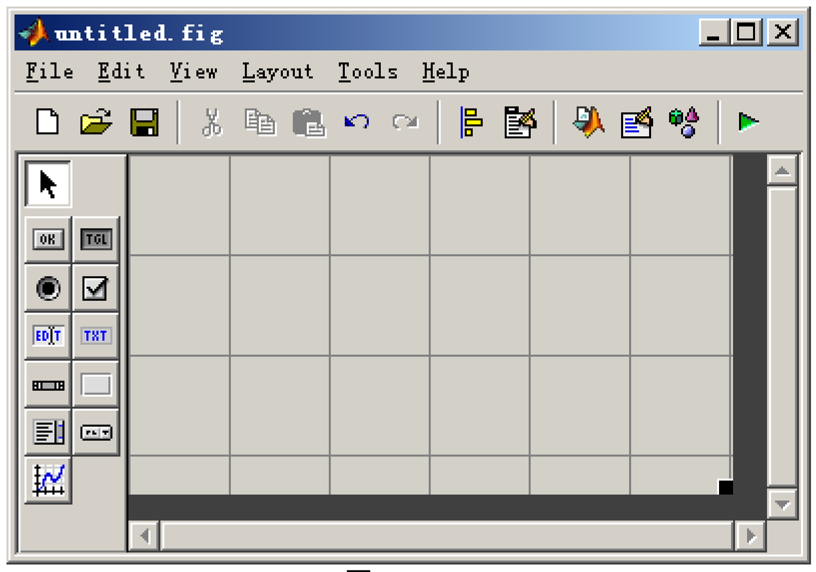


图 10.6.1-2

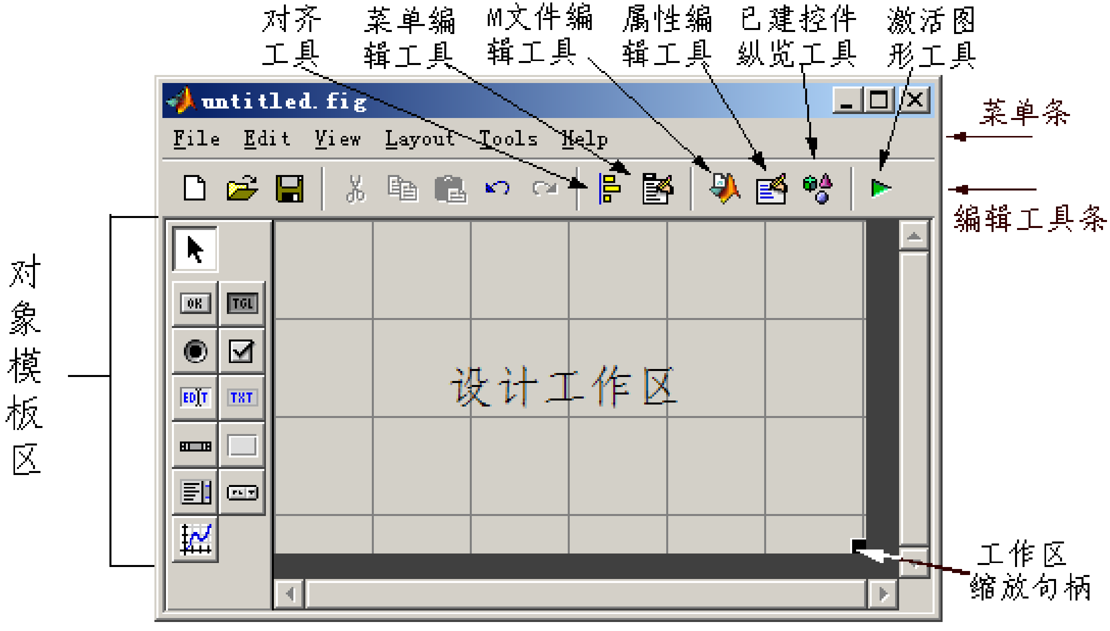


图 10.6.1-3

10.6.2 交互式用户界面设计工具应用示例

本节将以示例形式展开。

【例 10.6.2-1】使用 guide 来创建一个如图 10.6.2-1 所示的图形用户界面。该界面具有如下功能：

(1) 在编辑框中，可输入表示阻尼比的标量或“行数组”数值，并在按【Enter】键后，在轴上画出相应的蓝色曲线。坐标范围：X 轴 [0, 15]；Y 轴 [0, 2]。

(2) 在点击【Grid on】或【Grid off】键时，在轴上画出或删除“分格线”；缺省时，无分格线。

(3) 在菜单【Options】下，有 2 个下拉菜单项【Box on】和【Box off】；缺省时为 Box off 状态。

(4) 所设计的界面和其上图形对象、控件对象都按比例缩放。

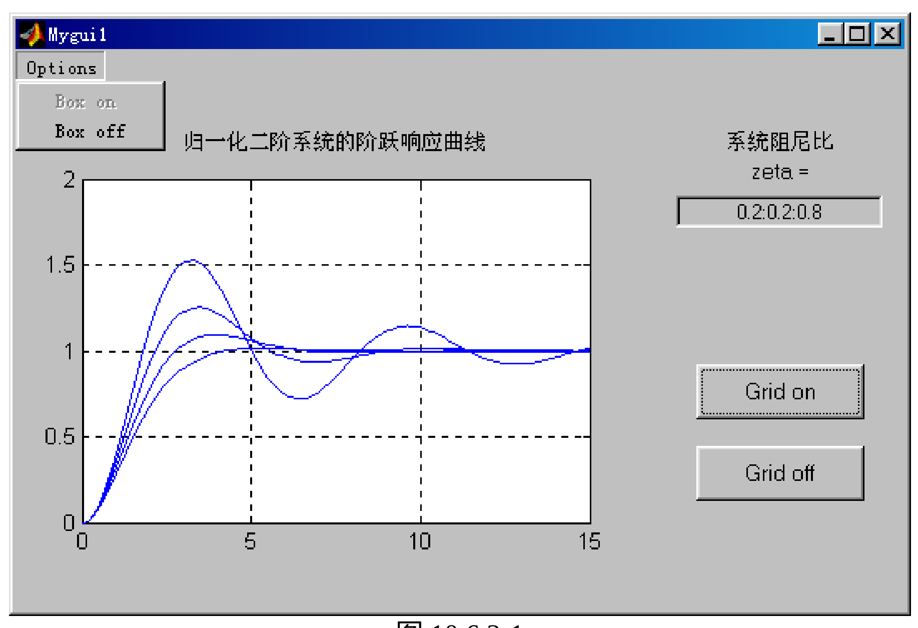


图 10.6.2-1

（1）步骤一：

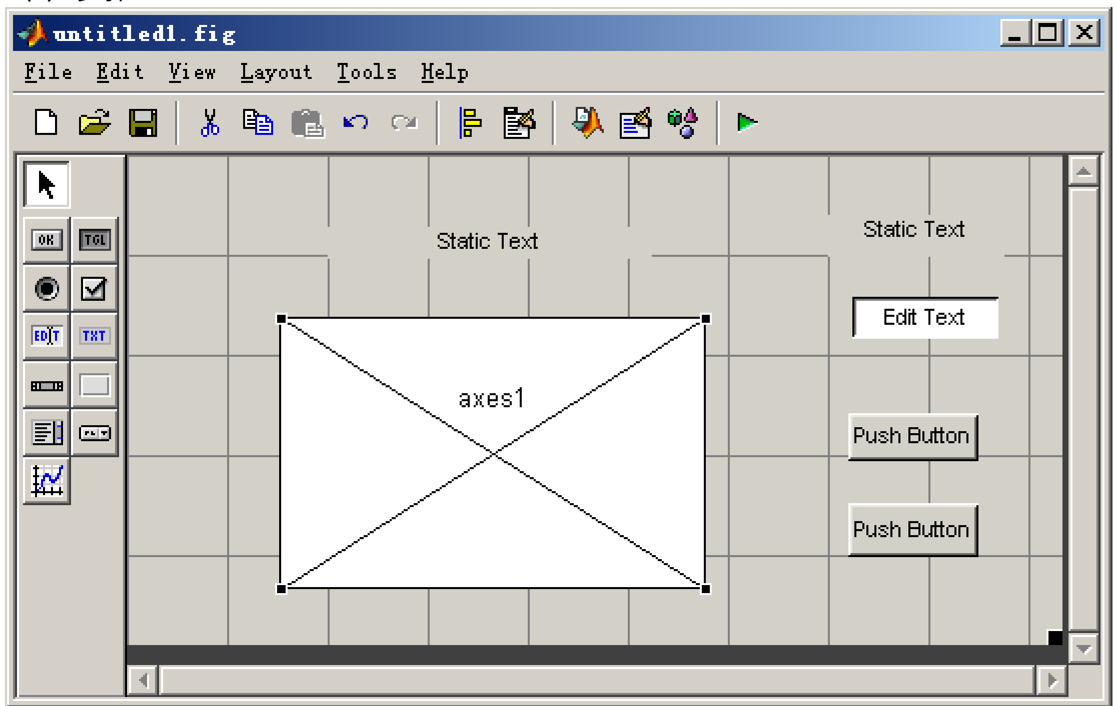


图 10.6.2-2

（2）步骤二：

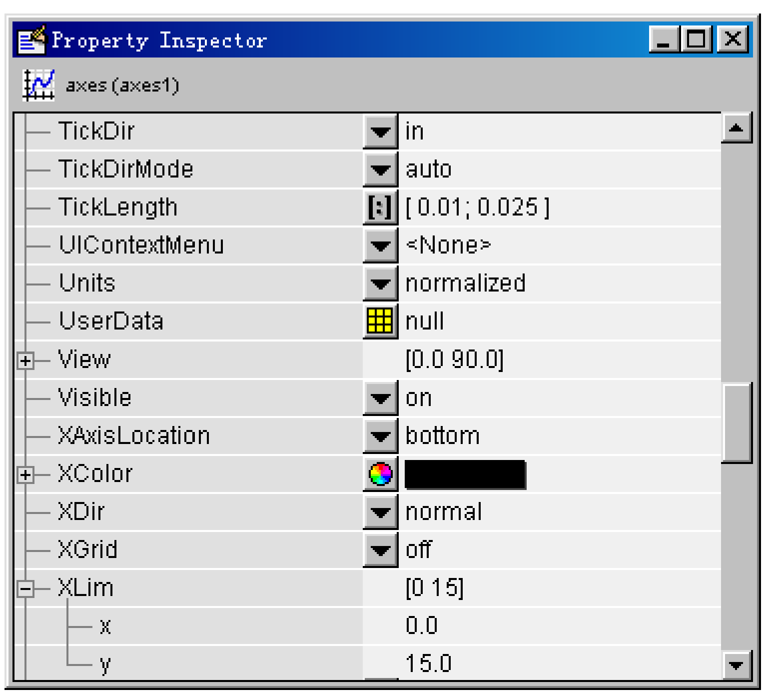


图 10.6.2-3

（3）步骤三：

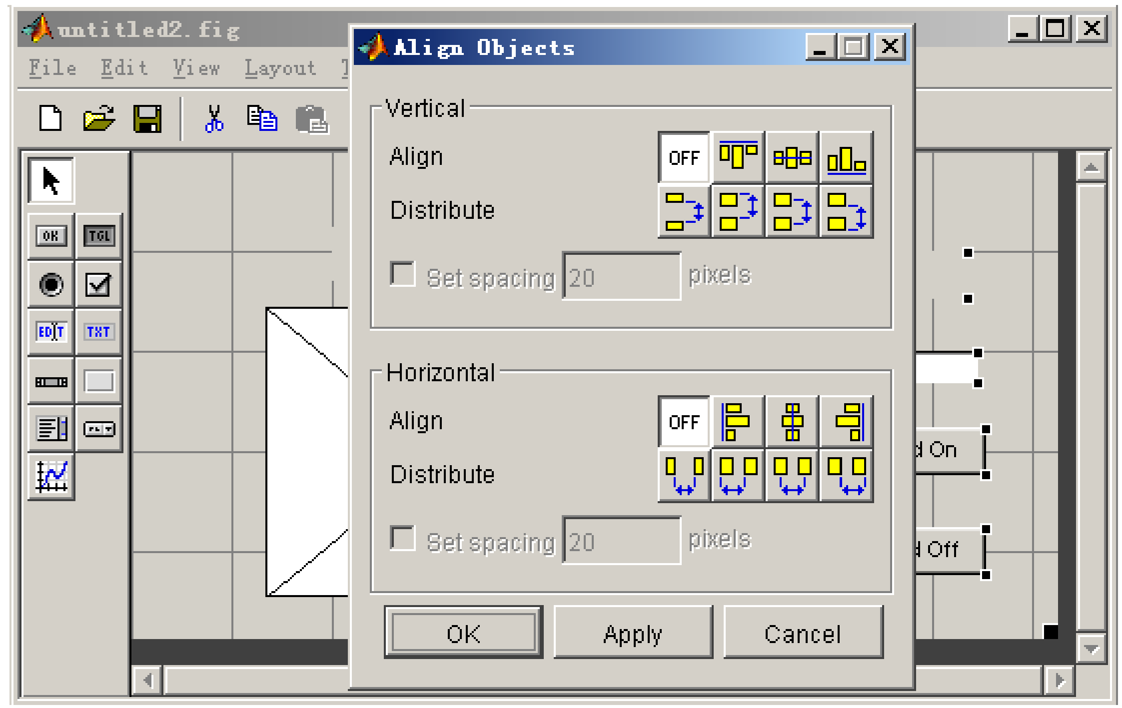


图 10.6.2-4

（4）步骤四：

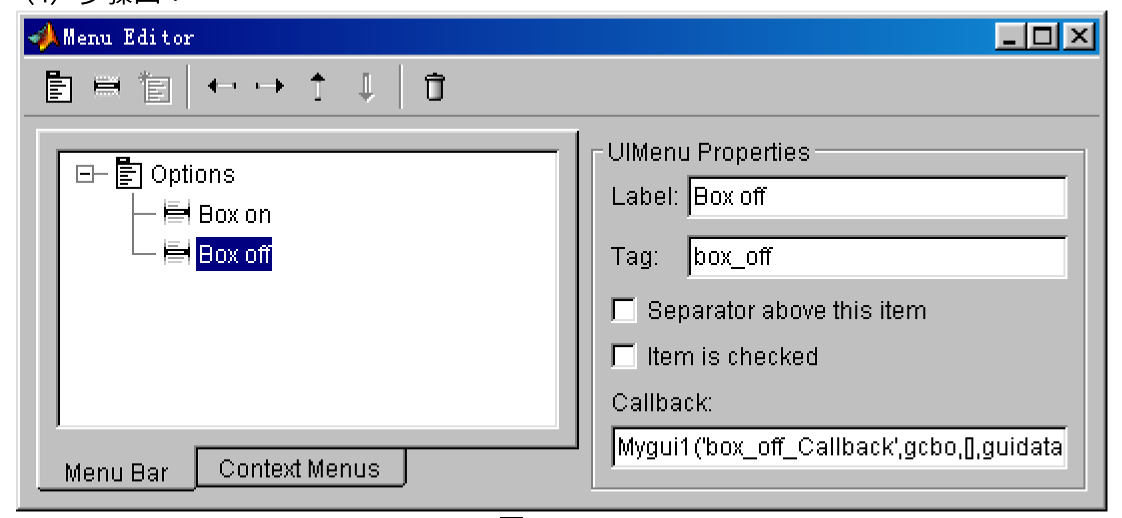


图 10.6.2-5

（5）步骤五：

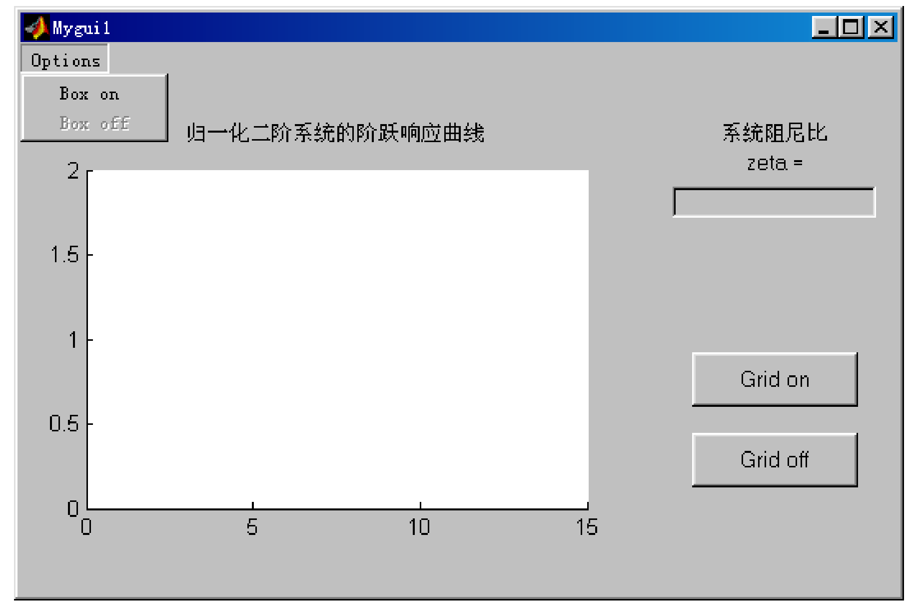
```
[mygui1.m]
function varargout = Mygui1(varargin)
% MYGUI1 Application M-file for Mygui1.fig
%    FIG = MYGUI1 launch Mygui1 GUI.
%    MYGUI1('callback_name', ...) invoke the named callback.
% Last Modified by GUIDE v2.0 15-Jun-2002 16:12:52
if nargin == 0  % LAUNCH GUI
    fig = openfig(mfilename,'reuse');
    % Use system color scheme for figure:
    set(fig,'Color',get(0,'defaultUicontrolBackgroundColor'));
    % Generate a structure of handles to pass to callbacks, and store it.
    handles = guihandles(fig);
```

```
        guidata(fig, handles);
        set(handles.box_off,'enable','off')
        if nargout > 0
            varargout{1} = fig;
        end
elseif ischar(varargin{1})      % INVOKE NAMED SUBFUNCTION OR CALLBACK
        try
            if (nargout)
                [varargout{1:nargout}] = feval(varargin{:}); % FEVAL switchyard
            else
                feval(varargin{:}); % FEVAL switchyard
            end
        catch
            disp(lasterr);
        end

end

% --------------------------------------------------------------------
function varargout = GridOff_push_Callback(h, eventdata, handles, varargin)
grid off
% --------------------------------------------------------------------
function varargout = GridOn_push_Callback(h, eventdata, handles, varargin)
grid on
% --------------------------------------------------------------------
function varargout = zeta_edit_Callback(h, eventdata, handles, varargin)
z=str2num(get(handles.zeta_edit,'String'));
t=0:0.1:15;
cla
for k=1:length(z)
   y(:,k)=step(1,[1, 2*z(k), 1], t);
   line(t,y(:,k));
end
% --------------------------------------------------------------------
function varargout = options_Callback(h, eventdata, handles, varargin)
% --------------------------------------------------------------------
function varargout = box_on_Callback(h, eventdata, handles, varargin)
box on
set(handles.box_on,'enable','off')
set(handles.box_off,'enable','on')
% --------------------------------------------------------------------
function varargout = box_off_Callback(h, eventdata, handles, varargin)
box off
set(handles.box_off,'enable','off')
set(handles.box_on,'enable','on')
```

（6）步骤六：


Mygui1
Options
Box on
Box off
归一化二阶系统的阶跃响应曲线
系统阻尼比
zeta =
2
1.5
1
0.5
0
0
5
10
15
Grid on
Grid off
图 10.6.2-6

第十一章 MATLAB 编译器

几乎所有使用过 MATLAB 的科技人员，无不为该软件的简洁、便捷和功能之强大和可靠所震撼，同时也对 MATLAB 产生了新的期望：一，希望程序能运行得更快；二，希望获得可摆脱 MATLAB 环境而独立运行的可执行软件。

由于 MATLAB6.5 版采用的编译器（Compiler）已经全面升级，本章内容是在 Compiler3.0 基础上全部重写的。

11.1编译器概述

11.1.1编译器的功能

11.1.2编译器的性能改进

11.1.3编译器的局限性

11.1.4把脚本文件改写为函数文件

【例 11.1.4-1】有一个绘圆的 M 脚本文件 circle.m 如下。希望获得一个 MEX 绘圆程序。

（1）

[circle.m]

```
clf;r=2;t=0:pi/100:2*pi;x=r*exp(i*t);
plot(x,'r*');axis('square')
```

（2）

```
mcc -x circle
??? Error: File "circle" is a script M-file and cannot be compiled with the current Compiler.

Error in ==> D:\MATLAB6P5\toolbox\compiler\mcc.dll
```

（3）

[circle_f.m]:

```
function circle_f(r)
clf;t=0:pi/100:2*pi;x=r*exp(i*t);
plot(x,'r*');axis('square')
```

（4）

```
mcc -x circle_f
```

（5）

```
circle_f(0.5)
which circle_f
d:\mywork\circle_f.dll
```

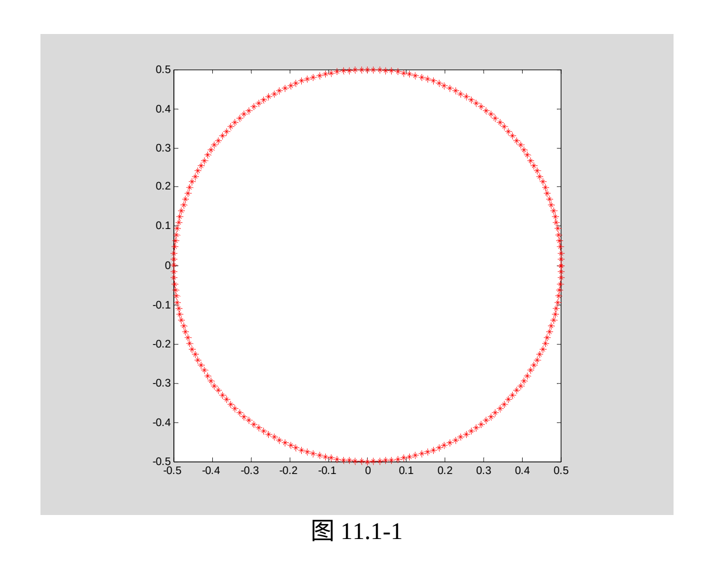


图 11.1-1

11.2编译器的安装和配置

11.2.1配置 MATLAB 编译器的前提准备

（1）

（2）

11.2.2为产生 MEX 文件进行预配置

11.2.2.1 对 MATLAB 编译器应用程序 mex 的设置

11.2.2.2 配置正确性的验证

（1）

```
cd d:\mywork
mex my_yprime.c
my_yprime(1,1:4)
which my_yprime
ans =
    2.0000    8.9685    4.0000   -1.0947
d:\mywork\my_yprime.dll
```

（2）

```
mcc -x my_yprime_m                                              %<1>
my_yprime_m(1,1:4)
which my_yprime_m
ans =
    2.0000
    8.9685
    4.0000
   -1.0947
d:\mywork\my_yprime_m.dll
```

（3）

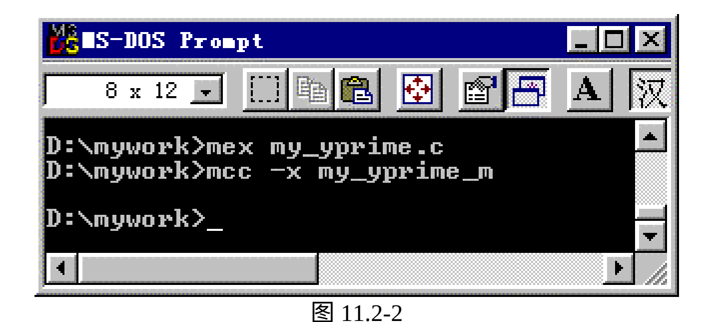


图 11.2-2

11.2.3为产生独立外部应用程序进行预配置

11.2.3.1 对 MATLAB 编译器 mbuild 应用程序的设置

11.2.3.2 配置正确性的验证

（1）

```
cd d:\mywork
mbuild my_ex1.c
```



图 11.2-3

（2）

```
mcc -p my_hello.m                                          %<1>
```

```
D:\mywork>
D:\mywork>my_hello
Hello, World

D:\mywork>
```

图 11.2-4

（3）

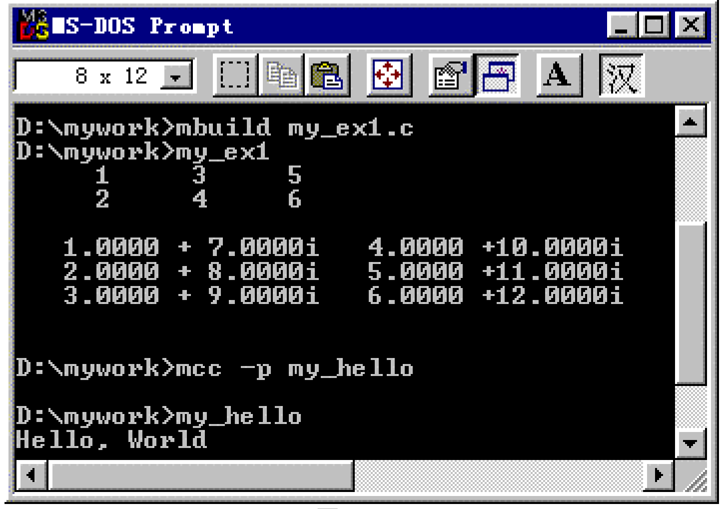


图 11.2-5

11.3MATLAB 编译器使用入门

11.3.1由 M 文件创建 C MEX 文件的入门算例

【例 11.3.1-1】先编写 M 文件，然后生成相应的 MEX 文件。该文件用以判断方阵是否奇异。

```
cd d:\mywork
```

（1）

[exm1.m]

```
function y=exm1(A)
[m,n]=size(A);
if m~=n;
    error('An input matrix should be n-by-n.')
end
r=rank(A);
if r==m
    disp('This matrix is nonsigular')
else
    disp('This matrix is sigular')
end
```

（2）

（3）

```
mcc -x exm1
```

（4）

```
A=[1,0,1;2,1,0;4,1,4]
exm1(A)
which exm1
A =
     1     0     1
     2     1     0
     4     1     4
This matrix is nonsigular
d:\mywork\exm1.dll
```

11.3.2由 M 文件创建外部应用程序的入门算例

【例 11.3.2-1】建立一个脱离 MATLAB 环境，可独立运行的外部程序。该程序的功能是：对于给定矩阵 A，如果存在 S 使得 $S^{-1}AS=\Lambda$，则要求出一个 S，否则给出信息说明所给的矩阵 A 不能对角化。

（1）

[exm2.m]

```
function exm2
A=[4,0,0;0,3,1;0,1,3];
S=exm2_f(A)
```

[exm2_f.m]

```
function S=exm2_f(A)
[m,n]=size(A);
if m~=n
    error('输入矩阵应是方阵！');
end;
e=eig(A);
%
same=0;
for i=1:m-1
   for j=(i+1):m
      if e(j)==e(i)
         same=1;
      end
   end
end
%
if any(any((A'-A)))&(same==1)
   error('矩阵无法对角化！');
end
[v,d]=eig(A);
S=v;
```

（2）

```
exm2
S =
         0         0    1.0000
   -0.7071    0.7071         0
    0.7071    0.7071         0
```

（3）

```
mcc -m exm2 exm2_f
```

（4）



图 11.3-1

11.4编译指令 mcc 简介

11.4.1 mcc 的基本调用格式

11.4.2mcc 的选项标志

11.4.2.1 编译器（2.2 版）选项简介

11.4.2.2 在命令行中指定选项标志

【例 11.4.2.2-1】假设当前目录上存在一个文件 exm3.m ，现要求利用 M 编译器将它转换为 C++语言的源码文件，并要求将原 M 文件中那注释区的内容作为所得 C++源码文件的注释。

```
mcc -t -L Cpp -A annotation:comments exm3
```

11.4.2.3 设置缺省选项

【例 11.4.2.3-1】假设已在<WINDOWS>\Application Data\MathWorks\MATLAB\R13 目录下创建了文本文件 mccstartup（请注意：该文件不可带扩展名），该文件的内容包括若干编译指令的选项。现要求利用该文件，从 exm3.m 出发得到 C++语言的源文件。

```
cd c:\ WINDOWS\Applic~1\MathWorks\MATLAB\R13
%
type C:\WINDOWS\Applic~1\MathWorks\MATLAB\R13\mccstartup
-t -L C -A annotation:none

!copy mccstartup h:\mywork\temp
!copy exm3.m h:\mywork\temp
!del mccstartup,
!del exm3*
dir mccstartup
%
cd d:\mywork
dir mccstartup
dir exm3*
        1 file(s) copied
        1 file(s) copied
```

```
mccstartup not found.
exm3* not found.

!copy h:\mywork\temp\exm3.m d:\mywork
mcc  -L Cpp exm3                                              %<1>
        1 file(s) copied
??? Error: The options specified will not generate any output files.
Please use one of the following options to generate an executable output
file:

  -x (generates a MEX-file executable using C)
  -m (generates a stand-alone executable using C)
  -p (generates a stand-alone executable using C++)
  -S (generates a Simulink MEX S-function using C)
  -B sgl (generates a stand-alone graphics library executable using C
(requires the SGL))
  -B sglcpp (generates a stand-alone graphics library executable using
C++ (requires the SGL))
  -B pcode (generates a MATLAB P-code file)

Or type mcc -? for more usage information.

Error in ==> D:\MATLAB6P5\toolbox\compiler\mcc.dll

!copy  h:\mywork\temp\mccstartup  C:\WINDOWS\Applic~1\MathWorks\MATLAB
\R13
mcc -L Cpp exm3                                               %<2>
dir exm3*                                                     %<3>
!del mccstartup
        1 file(s) copied
exm3.cpp exm3.hpp exm3.m
```

11.5编译文件的性能优化

11.5.1优化数组

【例 11.5.1-1】优化标量。假设有以下文件，要求对之编译，比较得到 MEX 文件的性能。

[foo.m]

```
function y = foo(x)
y = 2*pi*x;
```

分别运行以下编译指令：

```
mcc -O none -x foo
tic;foo(1:10);toc
elapsed_time =
   0.5500

mcc -O none  -O fold_scalar_mxarrays:on -x foo
tic;foo(1:10);toc
elapsed_time =
   0.1100
```

【例 11.5.1-2】而对于非标量的情况，相应的可激活优化选项 fold_non_scalar_mxarrays。优化编译以下文件：

[test.m]

```
function y = test
y = [ 1 0; 0 1] * [ pi pi/2; -pi -pi/2 ];
```

分别运行以下编译指令：

```
mcc -O none -x test
tic;test;toc
elapsed_time =
    0.8200
mcc -O none  -O fold_non_scalar_mxarrays:on -x test
tic;test;toc
elapsed_time =
     0
```

11.5.2优化循环

【例 11.5.2-1】激活选项 array_indexing，可以改善简单的一维或二维数组的索引性能。如果在编译时关闭该选项，编译器将采用通用的索引函数来索引这些简单的数组。

[test2.m]

```
function y = test2(x,i1,i2);
y = x(i1,i2);
```

分别运行以下指令：

```
A=magic(4);
mcc -O none -x test2
tic;test2(A,3,4);toc
elapsed_time =
  0.2200

mcc -O none  -O fold_non_scalar_mxarrays:on -x test2
tic;test2(A,3,4);toc
elapsed_time =
     0
```

【例 11.5.2-2】激活选项 optimize_integer_for_loops，可简化循环。当循环变量的初值和步长均为整数时，编译器将采用 C/C++的整型变量，而非 MATLAB 的数组变量。（建议在调试程序时，不要激活该优化选项。）

[test3.m]

```
function y=test3(x)
for i = 1:length(x)-1
   x(i) = x(i) + x(i+1);
end
y=x;
```

分别运行以下指令，对比各自编译得到的 MEX 文件的性能：

```
mcc -O none -x test3
tic;test3(1:100);toc
elapsed_time =
0.4400

mcc -O none  -O fold_non_scalar_mxarrays:on -x test3
tic;test3(1:100);toc
elapsed_time =
     0
```

11.5.3优化条件语句

【例 11.5.3-1】当条件语句中的两个运算量都是标量整数时，激活选项 optimize_conditionals，编译器将优化该条件语句：采用 C 的条件运算取代原 MATLAB 的条件运算。MATLAB 能

够识别出 nargin、nargout、for 语句的循环变量以及所有整数标量。

[tset4.m]

```
function test4(a,b,c,d)
   if (nargin < 4)
      d = 0.0;
   end
```

运行以下指令，对之进行不同的编译，并比较各自得到的 MEX 文件性能：

```
mcc -O none -x test4
tic;test4(1,3,4);toc

mcc -O none  -O fold_non_scalar_mxarrays:on -x test4
tic;test4(1,3,4);toc
```

11.6创建独立的外部应用程序

11.6.1独立外部程序的工作特点和创建过程

11.6.1.1　独立外部程序与 MEX 文件的不同工作特点

11.6.1.2　独立外部程序创建过程说明

11.6.2关于指令 mbuild

11.6.3借助编译指令 mcc 创建独立应用程序

11.6.3.1　创建独立应用程序时 mcc 的使用格式和常用选项标志

11.6.3.2　由全 M 源文件产生 EXE 应用程序

【例 11.6.3.2-1】创建一个适应“超定”、“恰定”、“欠定”线性方程求解的示例性应用程序。

（1）

[LLS.m]

```
function LLS()
Ae=5;                                                          %<2>
Av=2;                                                          %<3>
[A,b]=LLSDATA(Ae,Av);
x=A\b;
%
S='恰定';
if Ae>Av
   S='超定';
elseif Ae<Av
   S='欠定';
end
disp(['用外部独立程序求如下' S '方程 Ax=b 的解，其中'])
cs=blanks(Ae)';
ns=fix(Ae/2);
As=cs;As(ns)='A';
bs=cs;bs(ns)='b';
```

```
es=cs;es(ns)='=';
disp([As cs es cs num2str(A) cs cs cs cs cs bs cs es cs num2str(b)])
disp('方程的解 ')
nxs=fix(Av/2);
cxs=blanks(Av)';
xs=cxs;xs(nxs)='x';
exs=cxs;exs(nxs)='=';
disp([xs cxs exs cxs num2str(x)])
```

[LLSDATA.m]

```
function [A,b]=LLSDATA(Ae,Av)
n=max(Ae,Av);
WA=magic(n);
A=WA(:,1:Av);
if n>Ae
   A=WA(1:Ae,:);
end
b=ones(Ae,1);
```

(2)



图 11.6-1

(3)

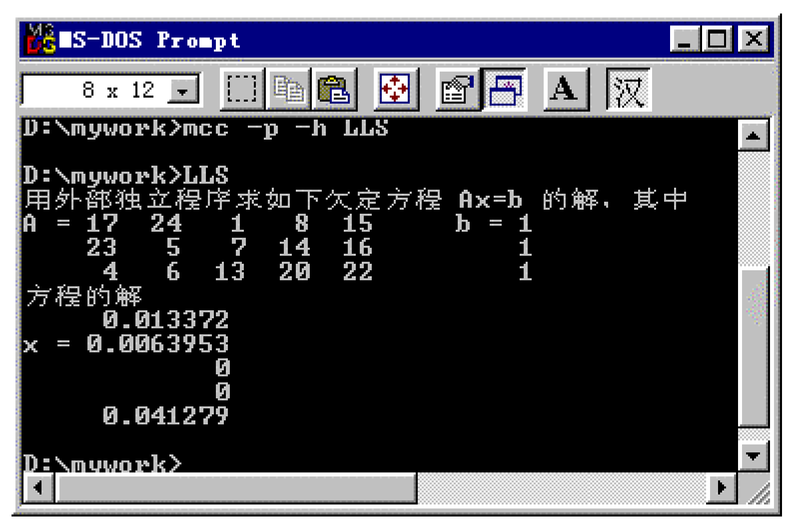


图 11.6-2

11.6.3.3 由包含绘图指令的 M 文件创建独立应用程序

【例 11.6.3.3-1】考虑本文在 7.4.3.1 节中给出的模仿卫星返回地球时运动轨迹图的【例 7.4.3.1-2】中的代码。假设其中最后一行语句中的彗尾长要求用户输入，据此对原组代码修改，并将该组代码构成为一个 M 函数文件，如下：

[exm110633_1.m]

```
function exm110633_1
shg;R0=1;
a=12*R0;b=9*R0;T0=2*pi;
T=5*T0;dt=pi/100;t=[0:dt:T]';
f=sqrt(a^2-b^2);
th=12.5*pi/180;
E=exp(-t/20);
x=E.*(a*cos(t)-f);y=E.*(b*cos(th)*sin(t));z=E.*(b*sin(th)*sin(t));
plot3(x,y,z,'g')
[X,Y,Z]=sphere(30);X=R0*X;Y=R0*Y;Z=R0*Z;
grid on,hold on,surf(X,Y,Z),shading  interp
x1=-18*R0;x2=6*R0;y1=-12*R0;y2=12*R0;z1=-6*R0;z2=6*R0;
axis([x1 x2 y1 y2 z1 z2])
view([117 37]),
p=input('Please input the comet" length coeffient (the default value is 0.1): ');
if ~isempty(p),
    p=0.1;
end
comet3(x,y,z,p),hold off
```

```
mcc -B sgl  exm110633_1
mcc -B sglcpp  exm110633_1
```

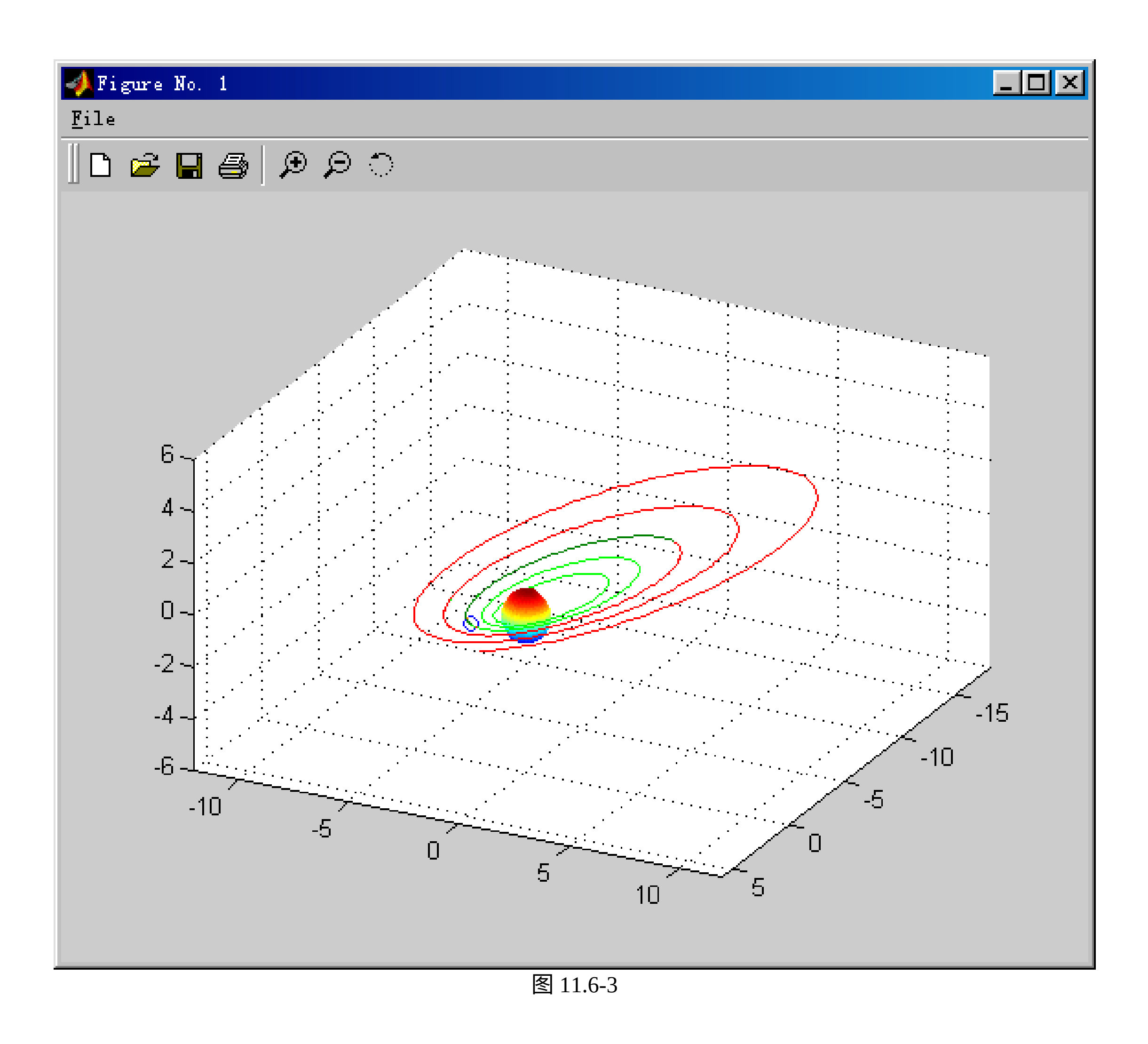


图 11.6-3

11.6.3.4　由含 feval 指令的 M 文件生成 EXE 文件

【例 11.6.3.4-1】要求生成一个可以计算方阵各种特征量的独立外部应用程序。

（1）

[mat_feat.m]

```
function mat_feat(f_name)                                         %<1>
disp('被分析矩阵')                                                %<2>
A=magic(4)
N=8;
n=size(f_name,2);
ff_name=[f_name blanks(N-n)];
if ff_name==['my_det' blanks(2)]
   disp('矩阵 A 的行列式值 = ')
elseif ff_name==['rank' blanks(4)]
   disp('矩阵 A 的秩 = ')
elseif ff_name==['norm' blanks(4)]
   disp('矩阵 A 的2-范数 = ')
elseif ff_name==['cond' blanks(4)]
   disp('矩阵 A 的条件数 = ')
elseif ff_name==['eig' blanks(5)]
   disp('矩阵 A 的特征值 = ')
elseif ff_name==['svd' blanks(5)]
```

```
    disp('矩阵 A 的奇异值 = ')
else
    disp('您输入的指令，或者不是本函数文件所能解决的，或是错误的！')
end
d=feval(f_name,A);
disp(d)
```

[my_det.m]

```
function d=my_det(A)
d=det(A);
```

（2）

[mainrank.m]

```
function mainrank
mat_feat('rank')
```

```
mcc -p mainrank
```

或者

```
mcc -p mainrank mat_feat
```

（3）

```
!mainrank
被分析矩阵
A =
        16          2          3         13
         5         11         10          8
         9          7          6         12
         4         14         15          1
矩阵 A 的秩 =
       3
```

（4）

[maindet.m]

```
function maindet
mat_feat('my_det')
```

```
mcc -p maindet my_det
```

或者

```
mcc -p maindet mat_feat my_det
```

```
!maindet
被分析矩阵

A =

    16     2     3    13
     5    11    10     8
     9     7     6    12
     4    14    15     1

矩阵 A 的行列式值 =
     0
```

【例 11.6.3.4-2】采用**编译注记法**生成一个可以计算方阵行列式值的独立外部应用程序。

（1）

```
%#function my_det
f_name='my_det';
```

（2）

```
mcc -p mat_feat
```

【例 11.6.3.4-3】当 feval 调用的是 MATLAB C++库中的函数时，可以采用更简单的 feval 输入宗量直接赋值法，实现 EXE 文件的创建。如创建一个计算方阵特征值的独立外部应用程序。

（1）

```
f_name='eig';
```

（2）

```
mcc -p mat_feat
```

（3）

```
!mat_feat
被分析矩阵
 A =
    16      2      3     13
     5     11     10      8
     9      7      6     12
     4     14     15      1

矩阵 A 的特征值 =
   34.0000
    8.9443
   -8.9443
   -0.0000
```

11.6.3.5 由含泛函指令的 M 文件生成 EXE 程序

【例 11.6.3.5-1】创建一个求一元函数 $y=\frac{1}{x}\sin(xe^{0.6x})$ 局部最小值的独立外部应用程序。

（1）

```
xx=0:0.01:5;
yy=SAA(xx);
plot(xx,yy)
x0=2;
x=fminsearch(@SAA,x0);
y=SAA(x);
plot(xx,yy,'b-',x,y,'r.','MarkerSize', 20),grid
```

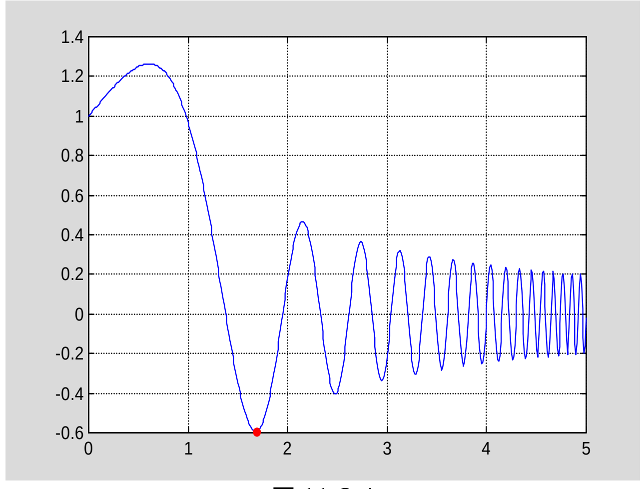


图 11.6-4

（2）

[fcpp.m]

```
function fcpp(fun)
%#function SAA
fun=@SAA;
xs=1;
x=fminsearch(fun,xs);
y=feval(fun,x);
disp(blanks(2)')
disp('独立外部程序 fcpp.exe 运行结果显示')
disp(blanks(2)')
ss=['sin(exp(0.6x))/x 函数在 x = ' num2str(xs) ' 附近达最小值的坐标为    '];
disp(ss);
disp(blanks(1)');
disp(['      (' num2str(x) ',' num2str(y) ')'])
```

[SAA.m]

```
function y=SAA(x)
x=x+(x==0)*eps;
y=sin(x.*exp(0.6*x))./x;
```

（3）

```
mcc -p fcpp SAA
```

（4）



图 11.6-5

11.6.3.6 由 C/C++源码和 M 源码文件混合生成 EXE 应用文件

【例 11.6.3.6-1】主程序为 C 源码文件，被调用程序为 M 文件。

（1）

[fileinc.c]

```
#include <stdio.h>
#include "matlab.h"
#include "templib.h"

int main(int argc, char **argv[])
{
    int n ;
    mxArray *r;
    mxArray *N;
    TemplibInitialize();
    n = 5;
    N=mxCreateDoubleMatrix(1,1,mxREAL);
    *mxGetPr(N)=n;
    r = mlfMrank(N);
    mlfPrintMatrix(r);
    mxDestroyArray(r);
    mxDestroyArray(N);
    TemplibTerminate();
    return 0;
}
```

[mrank.m]

```
function r=mrank(n)
r=zeros(n,1);
for k=1:n
   r(k)=rank(magic(k));
end
```

（2）

```
mcc -t -W lib:Templib -T link:exe -h mrank fileinc.c
mcc -t -W lib:Templib -T link:exe -h fileinc.c mrank
```

（3）

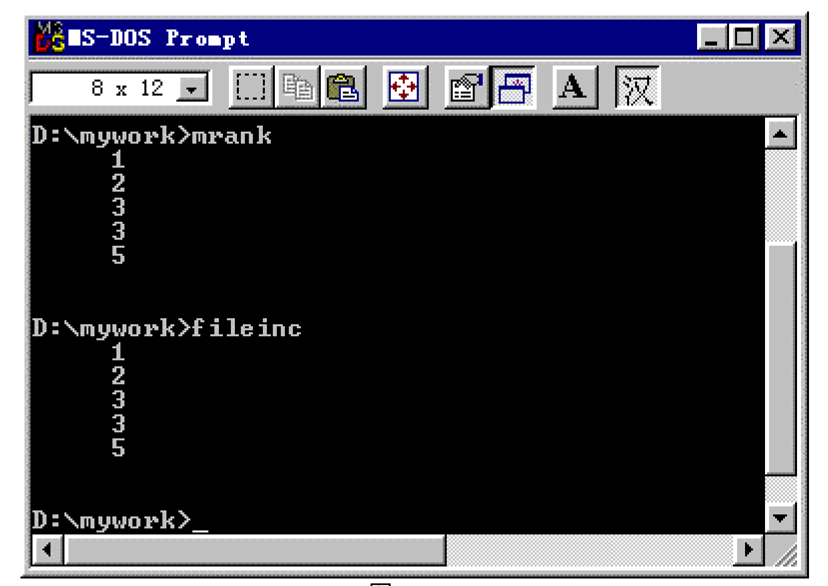

MS-DOS Prompt
8 x 12
D:\mywork>mrank
1
2
3
3
5
D:\mywork>fileinc
1
2
3
3
5
D:\mywork>_

图 11.6-6

第十二章 应用程序接口 API

前面章节主要叙述 MATLAB 自身的各种功能和使用方法。作为优秀软件，MATLAB 不仅自身功能强大、环境友善、能十分有效地处理各种科学和工程问题，而且具有极好的开放性。这开放性表现在两方面：一，MATLAB 适应各科学、专业研究的需要，提供了各种专业性的工具包；二，MATLAB 为实现与外部应用程序的“无缝”结合，提供了专门的应用程序接口 API。遵循本书“淡化专业，面向通用”的宗旨，本章将集中阐述 MATLAB 的应用程序接口。本章分五节，各节内容简述如下。

第 12.1 节集中介绍：如何编写 C MEX 源码程序，也就是如何为现有的 C 程序编写接口程序，使之成为 MATLAB 函数文件；运用这种技术，读者可以把积累的优秀 C 程序改造成可在 MATLAB 中方便调用的指令。

第 12.2 节用于演示：如何编写产生 MAT 数据文件的 C 源码程序。读者通过算例入门，就不难借助 MAT 文件实现 MATLAB 与外部应用程序的数据交换。

第 12.3 节围绕 MATLAB 引擎技术展开。借助这种技术，前台可以是各种外部应用程序编写的界面，而后台计算则可完全交由 MATLAB 进行。

第 12.4 节用三个算例来说明如何应用 ActiveX 实现 MATLAB 与外部应用程序的通信。在第一个算例中，MATLAB 用作为客户，服务器是 Excel。在后两个算例中，服务器是 MATLAB，而客户是 PowerPoint。由此产生的 PPT 文件，可以在放映过程中，实时地进行 MATLAB 调用。

第 12.5 节的内容是：如何借助 DDE 技术在 MATLAB 与其他外部程序间进行通信。该节的一个算例演示：VB 制作的界面如何借助 DDE 建立的对话通道调用服务器 MATLAB 进行计算和显示结果图形。而另一个算例则演示：MATLAB 如何以客户身份与服务器 Excel 建立 DDE“热连接”，使 MATLAB 图形实时地跟随电子表格数据的改变而变化。

值得指出：MATLAB6.0, 6.1 版用于 API 的（MEX、MAT、及引擎）库函数许多已经被废止。本章内容是根据 MATLAB6.5 编写的。

12.1 C 语言 MEX 文件的编写

12.1.1 关于 MEX 文件的一般性说明

12.1.2 MEX 文件中的 MATLAB 数据

```
A=['abcd';'1234';'ABCD']
```

```
A =
abcd
1234
ABCD
```

12.1.3 C 语言 MEX 文件源程序的构成

【例 12.1.3-1】列出具有相同运算功能（实现两个双精度实数标量加法）的 C++源码程序和 C++ MEX 源码程序；对 C++ MEX 源码程序进行编译链接；在 MATLAB 中调用生成的 DLL 文件。通过本例，从感性上认识：（A）一般 C 源码文件如何改写成具有约定格式的 C MEX 源码文件；（B）C MEX 源码文件的基本结构；（C）基本的编译链接方法；（D）DLL 文件的调用方法。

（1）

```
#include  <math.h>
void myplus(double y[],double x[],double z[])
{
      y[0]=x[0]+z[0];
      return;
}
```

（2）

```
[exm12013_1.cpp]
#include "mex.h"          //   <1>
//------------------------------------------------
void myplus(double y[],double x[],double z[])
{
      y[0]=x[0]+z[0];
}
//------------------------------------------------
void mexFunction(int nlhs, mxArray *plhs[], int nrhs, const mxArray *prhs[])             //<8>
{
      double *x,*y,*z;                                    //
      int mrows0,ncols0;                                  //
      int mrows1,ncols1;                                  //

      if (nrhs!=2)                                                                          //<13>
        mexErrMsgTxt("Two inputs required.");                                               //<16>
      else if (nlhs>1)                                                                      //<15>
        mexErrMsgTxt("Too many output arguments");                                          //<16>

      mrows0=mxGetM(prhs[0]);                                                               //<17>
      ncols0=mxGetN(prhs[0]);
      mrows1=mxGetM(prhs[1]);
      ncols1=mxGetN(prhs[1]);                                                               //<20>

      //
      if (!mxIsDouble(prhs[0])||mxIsComplex(prhs[0])||!(mrows0==1 && ncols0==1))    //<22>
```

```
    mexErrMsgTxt("Inputs must be all noncomplex scalar double.");
  //
  if (!mxIsDouble(prhs[1])||mxIsComplex(prhs[1])||!(mrows1==1 && ncols1==1))    //<25>
    mexErrMsgTxt("Inputs must be all noncomplex scalar double.");
  //
  if (mrows0!=mrows1||ncols0!=ncols1)                                           //<28>
    mexErrMsgTxt("Inputs must be same dimension.");                             //<29>

  //
  plhs[0]=mxCreateDoubleMatrix(mrows0,ncols0,mxREAL);                           //<31>

  x=mxGetPr(prhs[0]);                                                           //<32>
  z=mxGetPr(prhs[1]);                                                           //<33>
  y=mxGetPr(plhs[0]);                                                           //<34>

  myplus(y,x,z);
}
```

（**3**）

```
cd  D:\mywork
mex exm12013_1.cpp
dir exm12013_1.*
```

```
exm1213_1.cpp          exm1213_1.dll
```

（**4**）

```
a=0.111;b=0.222;
c=exm12013_1(a,b)
```

```
c =
    0.3330
```

12.1.4 C MEX 文件的执行流程

12.1.5 编写 C MEX 文件的常用库函数和示例

12.1.5.1 常用的 MEX 库函数

（**1**）

```
#include "mex.h"
void mexFunction(int nlhs, mxArray *plhs[], int nrhs, const mxArray *prhs[])
```

```
{
  /* 其他 C 源码……*/
}
```

（**2**）

```
#include "mex.h"
void mexErrMsgTxt(const char *error_msg);

void mexWarnMsgTxt(const char *warning_msg);
```

（**3**）

```
#include "mex.h"
int mexCallMATLAB(int nlhs, mxArray *plhs[], int nrhs,
                  mxArray *prhs[], const char *command_name);
```

（**4**）

```
#include "mex.h"
int mexEvalString(const char *command);
```

（**5**）

```
#include "mex.h"
mxArray *mexGetVariable(const char *workspace, const char *var_name);

int mexPutVariable(const char *workspace, const char *var_name, mxArray *array_ptr);
```

12.1.5.2 常用的 MX 函数

（**1**）

```
#include "matrix.h"
mxArray *mxCreateNumericMatrix(int m, int n, mxClassID class,
                               mxComplexity ComplexFlag);
```

（**2**）

```
#include "matrix.h"
int mxGetM(const mxArray *array_ptr);
int mxGetN(const mxArray *array_ptr);

void mxSetM(mxArray *array_ptr, int m);
void mxSetN(mxArray *array_ptr, int m);
```

（**3**）

```
#include "matrix.h"
double *mxGetPr(const mxArray *array_ptr);
double *mxGetPi(const mxArray *array_ptr);
```

```
void mxSetPr(mxArray *array_ptr, double *pr);
void mxSetPi(mxArray *array_ptr, double *pr);
```

（**4**）

```
#include "matrix.h"
#include <stdlib.h>
void *mxCalloc(size_t n, size_t size);
```

12.1.5.3 编程示例

【例 12.1.5.3-1】创建一个 C 语言 MEX 文件，实现对 MATLAB 两个“单行”字符串的合并。本例演示：（A）如何根据 MATLAB 约定的规则编写 C MEX 源码；（B）如何构成该文件的调用指令；（C）如何为 MEX 文件编写在线帮助文件。

（1）

```
#include "mex.h"                                                          //<1>
#include "string.h"                                                       //<2>

//-------------------------------------------------
void stringplus(char *input_buf0,char *input_buf1,char *output_buf)
{
        strcat(output_buf,input_buf0);
        strcat(output_buf,input_buf1);
}

//-------------------------------------------------
void mexFunction(int nlhs,mxArray *plhs[], int nrhs,const mxArray *prhs[])    //<10>
{
char *input_buf0,*input_buf1,*output_buf;
int buflen,buflen0,buflen1,status;

        if (nrhs!=2)                                                      //<13>
          mexErrMsgTxt("Two inputs inquired.");                           //<14>
        else  if (nlhs>1)                                                 //<15>
          mexErrMsgTxt("Too many output arguments.");                     //<16>

        if (mxIsChar(prhs[0])!=1||mxIsChar(prhs[1])!=1)                   //<17>
          mexErrMsgTxt("Inputs must be a string.");
        if (mxGetM(prhs[0])!=1||mxGetM(prhs[1])!=1)                       //<19>
          mexErrMsgTxt("Inputs must be a row vector.");

        buflen0=(mxGetM(prhs[0])*mxGetN(prhs[0]))+1;                      //<21>
        buflen1=(mxGetM(prhs[1])*mxGetN(prhs[1]))+1;                      //<22>
        buflen=buflen0+buflen1-1;
```

```
    input_buf0=(char *)mxCalloc(buflen0,sizeof(char));


    input_buf1=(char *)mxCalloc(buflen1,sizeof(char));
    output_buf=(char *)mxCalloc(buflen,sizeof(char));

    //
    status=mxGetString(prhs[0],input_buf0,buflen0);                    //<30>
    if (status!=0)
      mexWarnMsgTxt("Not enough space,String is truncated.");

    //
    status=mxGetString(prhs[1],input_buf1,buflen1);                    //<34>
    if (status!=0)
      mexWarnMsgTxt("Not enough space,String is truncated.");

    stringplus(input_buf0,input_buf1,output_buf);

    //
    plhs[0]=mxCreateString(output_buf);                                //<39>
    return;
}
```

（**3**）

```
cd d:\mywork
mex exm120153_1.cpp
```

（**4**）

根据以上分析，就可以写出下列 exm120153_1.m 文件：

```
% exm120153_1.m        Two strings are concatenated into a larger string.
% Cstr=exm120153_1(Astr, Bstr)     把字符串 Astr 和 Bstr 水平串联
%        Astr          被串联的“单行”字符串
%        Bstr          被串联的“单行”字符串
%        Cstr          由 Astr 在前，Bstr 在后，串联而成的字符串。

%  2002 年 11 月编写
```

（**5**）

```
A='1234';B='abcd';
C=exm120153_1(A,B)

 C =
```

```
1234abcd
```

【例 12.1.5.3-2】用 C 语言编写 MEX 源码文件，在运行中实现对 MATLAB 函数的调用，画出了 $y = 1 - e^{-0.3t}\cos(2t)$ 曲线。本例演示：（A）如何在 MEX 文件中调用 MATLAB 的内建指令；（B）如何在 MEX 文件中调用用户的自编 M 文件。

（1）

```
#include "mex.h"
#define MAX 1000

//-----------------------------------------------
void fill( double *pr, int *pm, int *pn, int max )
{
    int i;
    *pm = max/2;
    *pn = 1;
    for (i=0; i < (*pm); i++)
      pr[i]=i*(4*3.14159/max);
}

//-----------------------------------------------
void mexFunction( int nlhs, mxArray *plhs[], int nrhs, const mxArray *prhs[] )
{
    int m, n, max=MAX;
    mxArray *rhs[1], *lhs[1];

    rhs[0] = mxCreateDoubleMatrix(max, 1, mxREAL);

    fill(mxGetPr(rhs[0]), &m, &n, MAX);

    mxSetM(rhs[0], m);
    mxSetN(rhs[0], n);

    mexCallMATLAB(1, lhs, 1, rhs, "mexzzy");
    mexCallMATLAB(0, NULL, 1, lhs, "plot");

    mxDestroyArray(rhs[0]);
    mxDestroyArray(lhs[0]);

    return;
}
```

（2）

```
cd d:\mywork
mex exm120153_2.cpp
```

（3）

```
exm120153_2
```

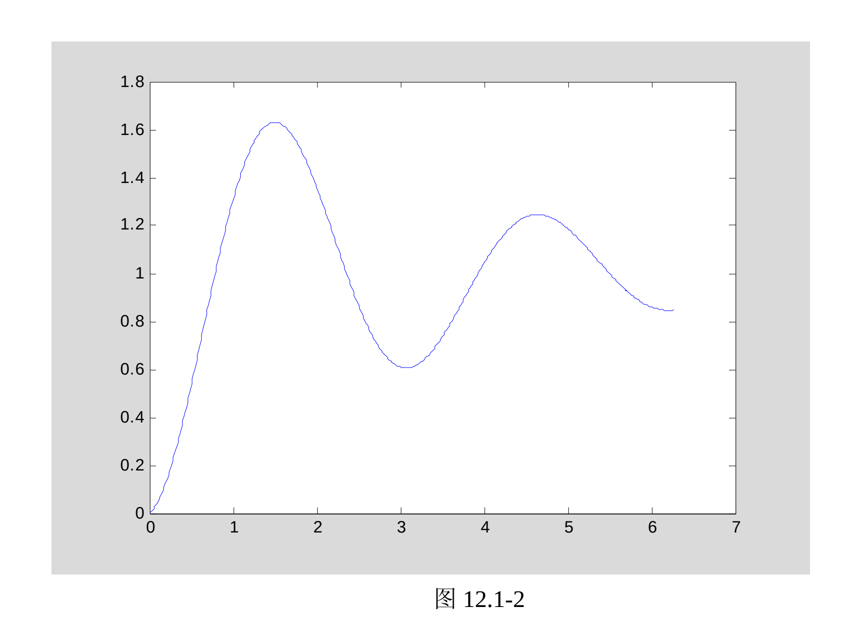


图 12.1-2

12.2 MAT 数据文件的应用

12.2.1 数据的输入输出方法

12.2.2 创建 MAT 文件的 C 源码程序的编写

【例 12.2.2-1】目标：用 C++编写一个可创建 MAT 文件的独立应用程序 exm12022_1.exe。通过该例演示：（A）可创建 MAT 文件的独立应用程序的编写步骤；（B）相应 C++源程序的基本格式；（C）相应 mx-函数和 C 指令的配合应用。（D）MAT 库函数 matClose, matGetArray, matOpen, matPutArray, matPutArrayAsGlobal 的使用方法；

（1）

```
#include <stdio.h>
#include "mat.h"
#include <string.h>
#define BUFSIZE 255

//--------------------------------------------------------
int create(const char *file)
{
	//
```

```
MATFile *pmat;
mxArray *pa1, *pa2, *pa3;
double data[9] = { 1.0, 4.0, 7.0, 2.0, 5.0, 8.0, 3.0, 6.0, 9.0 };
char str[BUFSIZE];
printf("Creating file %s...\n\n", file);
pmat = matOpen(file, "w");

if (pmat == NULL)
{
  printf("Error creating file %s\n", file);
  printf("(do you have write permission in this directory?)\n");
  return(1);
}

pa1 = mxCreateDoubleMatrix(3,3,mxREAL);
mxSetClassName(pa1, "LocalDouble");
pa2 = mxCreateDoubleMatrix(3,3,mxREAL);
mxSetClassName(pa2, "GlobalDouble");
memcpy((void *)(mxGetPr(pa2)), (void *)data, sizeof(data));


pa3 = mxCreateString("MATLAB: the language of technical computing");
mxSetClassName(pa3, "LocalString");
matPutVariable(pmat,"LocalDouble", pa1);
matPutVariableAsGlobal(pmat,"GlobalDouble", pa2);
matPutVariable(pmat,  "LocalString",pa3);


memcpy((void *)(mxGetPr(pa1)), (void *)data, sizeof(data));
matPutVariable(pmat,"LocalDouble",  pa1);

mxDestroyArray(pa1);
mxDestroyArray(pa2);
mxDestroyArray(pa3);

if (matClose(pmat) != 0)
{
  printf("Error closing file %s\n",file);
  return(1);
}

//
    pmat = matOpen(file, "r");
 if (pmat == NULL)
```

```
{
  printf("Error reopening file %s\n", file);
  return(1);
}

pa1 = matGetVariable(pmat, "LocalDouble");
        //
if (pa1 == NULL)
{
  printf("Error reading existing matrix LocalDouble\n");
  return(1);
}
if (mxGetNumberOfDimensions(pa1) != 2)
{
  printf("Error saving matrix: result does not have two dimensions\n");
  return(1);
}

pa2 = matGetVariable(pmat, "GlobalDouble");
        //
if (pa2 == NULL)
{
  printf("Error reading existing matrix GlobalDouble\n");
  return(1);
}
if (!(mxIsFromGlobalWS(pa2)))
{
  printf("Error saving global matrix: result is not global\n");
  return(1);
}

pa3 = matGetVariable(pmat, "LocalString");
        //
if (pa3 == NULL)
{
  printf("Error reading existing matrix LocalString\n");
  return(1);
}

mxGetString(pa3, str, 255);

if (strcmp(str, "MATLAB: the language of technical computing"))

{
```

```
        printf("Error saving string: result has incorrect contents\n");
        return(1);
      }

      mxDestroyArray(pa1);
      mxDestroyArray(pa2);
      mxDestroyArray(pa3);
      if (matClose(pmat) != 0)
      {
        printf("Error closing file %s\n",file);
        return(1);
      }
      printf("Done\n");
      return(0);
}

//--------------------------主程序-------------------------------
int main()
{
      int result;

      result = create("mattest.mat");
      return (result==0)?EXIT_SUCCESS:EXIT_FAILURE;

}
```

（**2**）

```
cd d:\mywork
mcc -p exm12022_1.cpp
```

（**3**）

```
clear
cd d:\mywork
!exm12022_1

Creating file mattest.mat...
Done

load mattest.mat
who

Your variables are:

GlobalDouble  LocalDouble   LocalString
```

```
GlobalDouble,LocalDouble,LocalString

GlobalDouble =
     1     2     3
     4     5     6
     7     8     9
LocalDouble =
     1     2     3
     4     5     6
     7     8     9
LocalString =
MATLAB: the language of technical computing
```

12.3 MATLAB 引擎技术的应用

12.3.1 MATLAB 引擎概念和功用

12.3.2 引擎库函数及 C 源码应用程序的编写

【例 12.3.2-1】用 C 语言编写调用 MATLAB 引擎计算三次多项式 x^3-2x+5 根的源程序。

（1）

```
#include <windows.h>
#include <stdlib.h>
#include <stdio.h>
#include "engine.h"

int PASCAL WinMain (HANDLE hInstance, HANDLE hPrevInstance,
                    LPSTR  lpszCmdLine, int   nCmdShow)
{
    Engine *ep;
    mxArray *P=NULL,*r=NULL;
    char buffer[301];
    double poly[4]={1,0,-2,5};

    if (!(ep=engOpen(NULL))) {
      fprintf(stderr,"\nCan't start MATLAB engine\n");
      return EXIT_FAILURE;}
```

```
    P=mxCreateDoubleMatrix(1,4,mxREAL);
    mxSetClassName(P,"p");
    memcpy((char *)mxGetPr(P),(char *)poly,4*sizeof(double));


    engPutVariable(ep,P);

    engOutputBuffer(ep,buffer,300);
    engEvalString(ep,"disp(['多项式',poly2str(p,'x'),'  的根']),r=roots(p)");



    MessageBox(NULL,buffer,"exm12032_1 展示 MATLAB 引擎的应用",MB_OK);



    engClose(ep);
    mxDestroyArray(P);
    return EXIT_SUCCESS;
}
```

（**2**）

```
cd d:\mywork
mex -f D:\MATLAB6p5\bin\win32\mexopts\msvc60engmatopts.bat exm12032_1.c
```

（**3**）

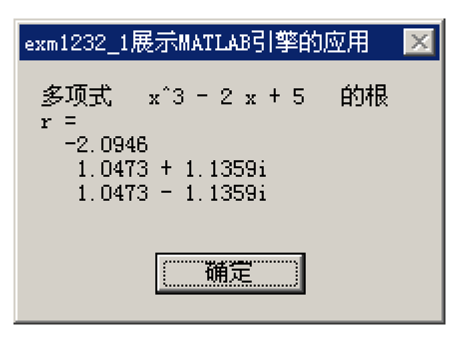


图 12.3-1

【例 12.3.2-2】MATLAB 引擎综合应用实例。本例演示：（A）MATLAB 引擎对用户自编 M 函数文件的调用；（B）借用 DOS 界面作为 MATLAB 的指令输入和结果发布窗。

（1）

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
```

```
#include "engine.h"
#define  BUFSIZE 512
int main()
{
  Engine *ep;
  mxArray *Pz = NULL, *result = NULL;
  char buffer[BUFSIZE];
  double zeta[4] = {0.2, 0.4, 0.8, 1.2 };         //MATLAB 环境外数据示例
        if (!(ep = engOpen("\0")))     //开启本地 MATLAB 引擎，如失败给出警告。<12>
        {
          fprintf(stderr, "\nCan't start MATLAB engine\n");
          return EXIT_FAILURE;
        }
//------------------------------------------------------------------------------------------------------------
//程序段 1：
//------------------------------------------------------------------------------------------------------------
        Pz = mxCreateDoubleMatrix(1, 4, mxREAL);
        mxSetClassName(Pz, "z");
        memcpy((void *)mxGetPr(Pz), (void *)zeta, sizeof(zeta));

        engPutVariable(ep, Pz);
        engEvalString(ep, "engzzy(z);");                                                        //<25>

        printf("按 Enter 键继续！ \n\n");
        fgetc(stdin);
        printf("程序段 1 运行已经结束。下面处于程序段 2 运行过程中！ \n");
        mxDestroyArray(Pz);
        engEvalString(ep, "close;");
//------------------------------------------------------------------------------------------------------------
//程序段 2：
//
//
//------------------------------------------------------------------------------------------------------------
        engOutputBuffer(ep, buffer, BUFSIZE);


        while (result == NULL) {
          char str[BUFSIZE];
          printf("注意：\n");
          printf("• 此界面上，可输入任何 MATLAB 指令。\n");
          printf("• 若想退出，请对 Exit 变量赋任何数值。\n");
          printf(">> ");
          fgets(str, BUFSIZE-1, stdin);
          engEvalString(ep, str);
```

```
        printf(" %s", buffer);
        if ((result = engGetArray(ep,"Exit")) == NULL)

            printf("可继续运行！ \n");
    }
    printf("运行结束!\n");
    mxDestroyArray(result);
    engClose(ep);
    return EXIT_SUCCESS;
}
```

（2）

```
cd d:\mywork
mex -f D:\MATLAB6p5\bin\win32\mexopts\msvc60engmatopts.bat exm12032_2.c
```

（3）

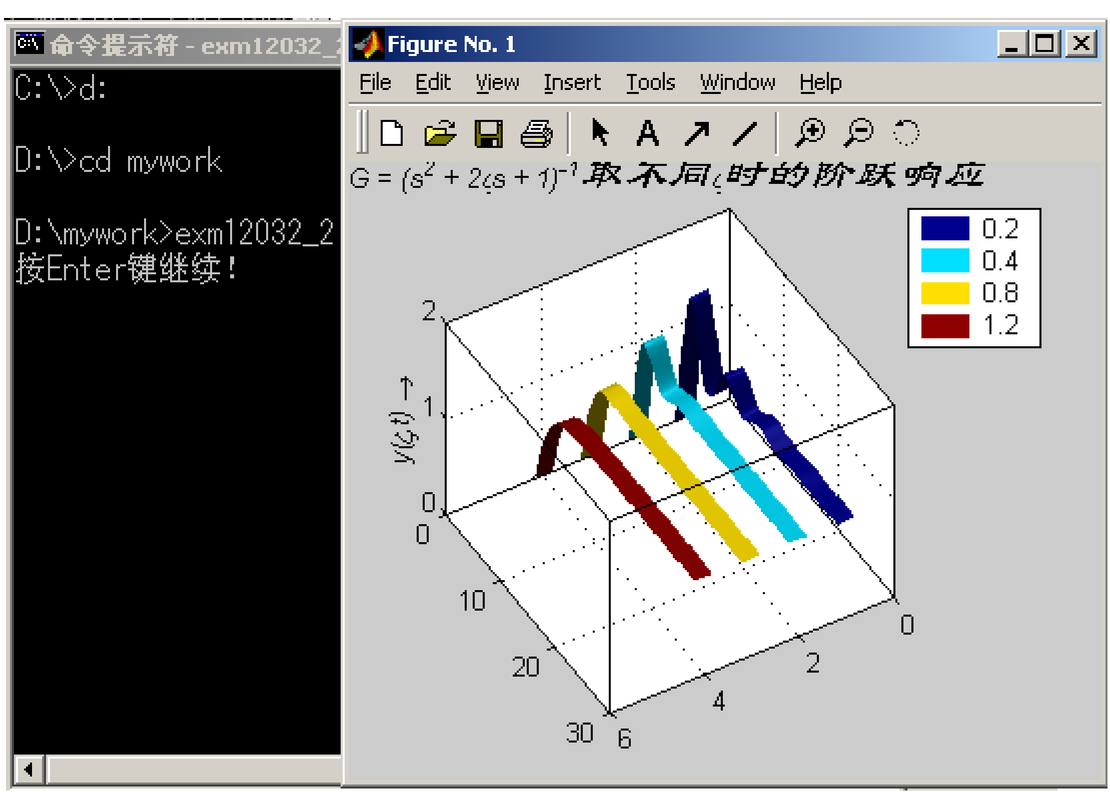


图 12.3-2



图 12.3-3

12.3.3 利用 VC++ 6.0 集成环境编写 MATLAB 引擎程序

【例 12.3.3-1】利用 VC++6.0 集成编程界面编写综合运用 MAT 数据文件和 MATLAB 引擎技术的（求矩阵的奇异值）C++源码程序。本例演示：（A）如何利用 VC++6.0 集成编程界面编写源码程序；（B）编译链接产生 EXE 文件所需的间夹文件。

（1）

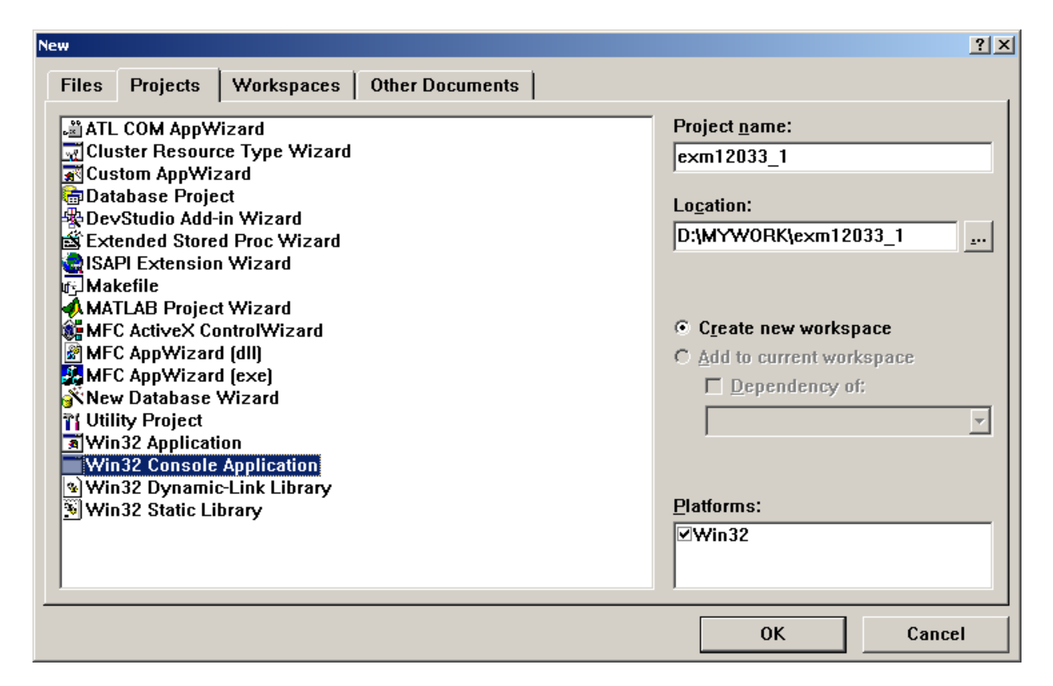

New
Files
Projects
Workspaces
Other Documents
ATL COM AppWizard
Cluster Resource Type Wizard
Custom AppWizard
Database Project
DevStudio Add-in Wizard
Extended Stored Proc Wizard
ISAPI Extension Wizard
Makefile
MATLAB Project Wizard
MFC ActiveX ControlWizard
MFC AppWizard (dll)
MFC AppWizard (exe)
New Database Wizard
Utility Project
Win32 Application
Win32 Console Application
Win32 Dynamic-Link Library
Win32 Static Library
Project name:
exm12033_1
Location:
D:\MYWORK\exm12033_1
Create new workspace
Add to current workspace
Dependency of:
Platforms:
Win32
OK
Cancel

图 12.3-4

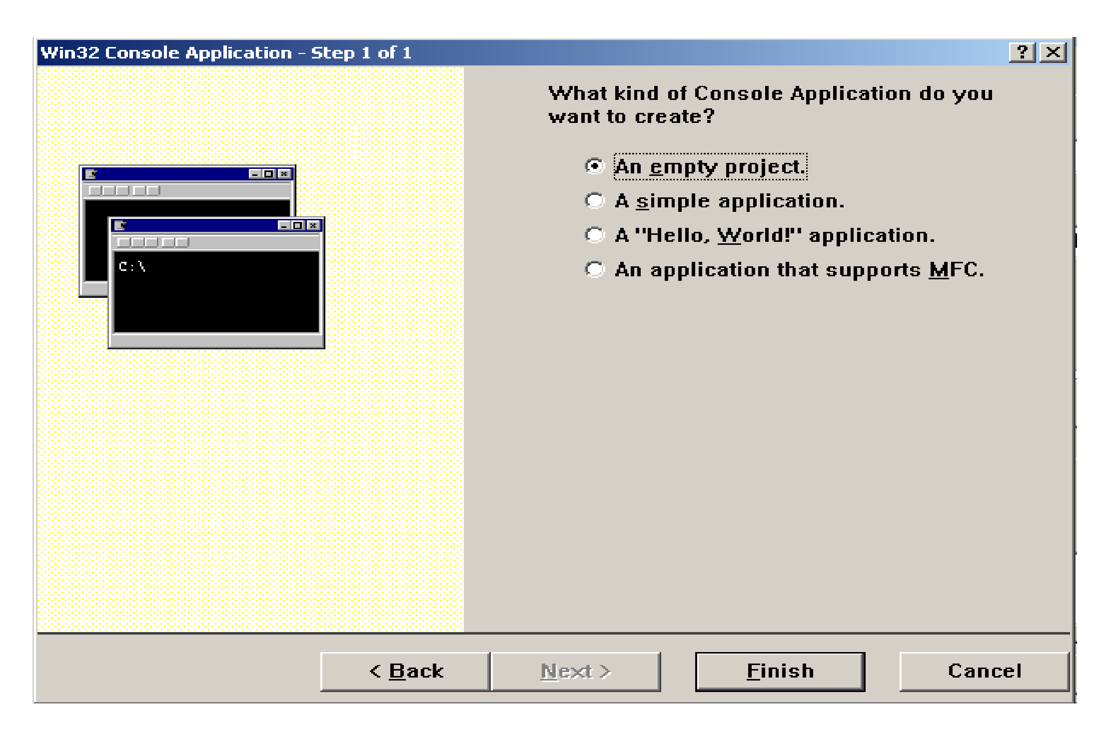

Win32 Console Application - Step 1 of 1
What kind of Console Application do you want to create?
An empty project.
A simple application.
A "Hello, World!" application.
An application that supports MFC.
< Back
Next >
Finish
Cancel

图 12.3-5

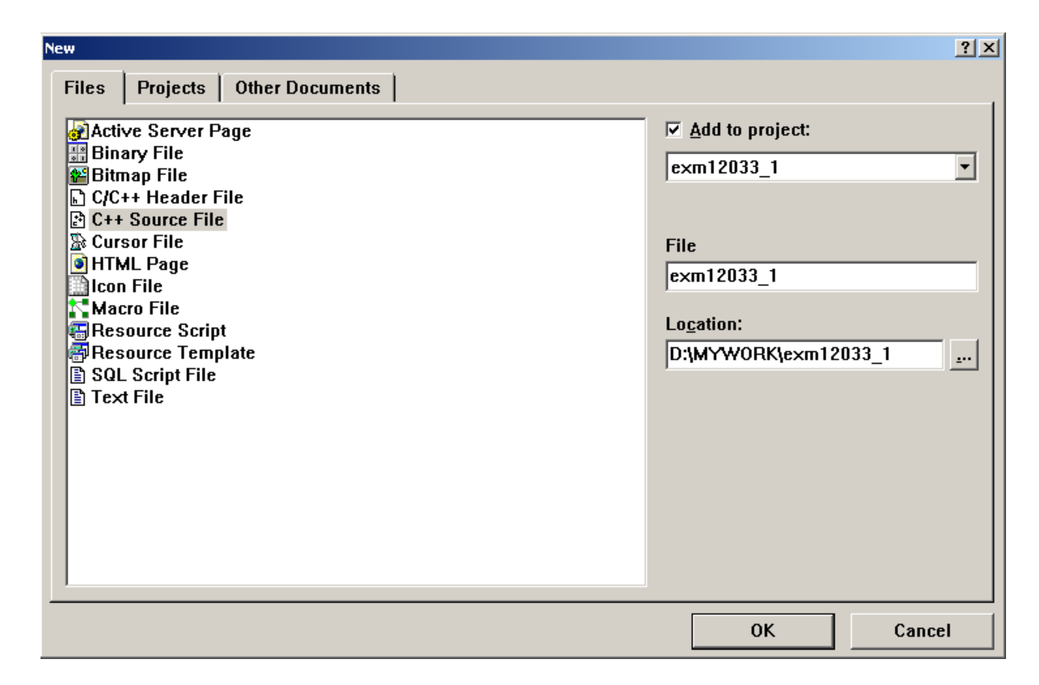


图 12.3-6

图 12.3-7

（2）

```
#include "engine.h"
#include "mat.h"
#include <iostream.h>
#include <windows.h>
#include <stdlib.h>
```

```
#include <iomanip.h>
void main()
{
	double * b,a[9]={1,4,7,2,5,8,3,6,9};
	const char *file="mymat.mat";
	mxArray *Ain,*Aout,*SV;
	Engine * ep;
	MATFile *mat;
//-----------------------------------------------------------
	mat=matOpen(file,"w");

	Ain = mxCreateDoubleMatrix(3,3,mxREAL);
	mxSetClassName(Ain,"z");
	memcpy((char *)mxGetPr(Ain),(char *)a,9*sizeof(double));


	matPutVariable(mat,"z",Ain);
	matClose(mat);
	mxDestroyArray(Ain);

//-------------------------------
	mat=matOpen(file,"r");

	Aout =matGetVariable(mat,"z");

	if(ep=engOpen(NULL))
	{
	  engPutVariable(ep,Aout);

	  engEvalString(ep,"sv=svd(z);");
	  SV=engGetVariable(ep,"sv");
	  b=mxGetPr(SV);
	  cout<<"奇异值为";
	  cout<<"\n";
	  for(int i=0;i<3;i++)
	  {
	       cout<<setw(16)<<b[i];
	  }
	  engClose(ep);
	  matClose(mat);
	  mxDestroyArray(Aout);
	  mxDestroyArray(SV);
	}
	else
```

```
        cout<<"Can't open matlab";
}
```

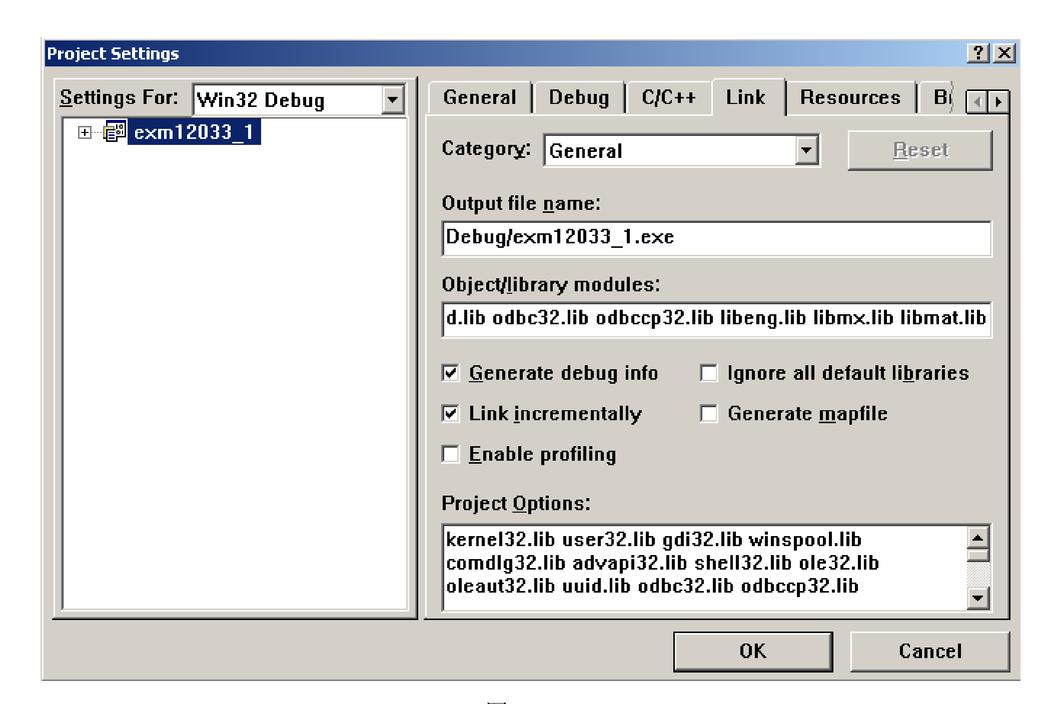


图 12.3-8

（3）

（4）

（5）

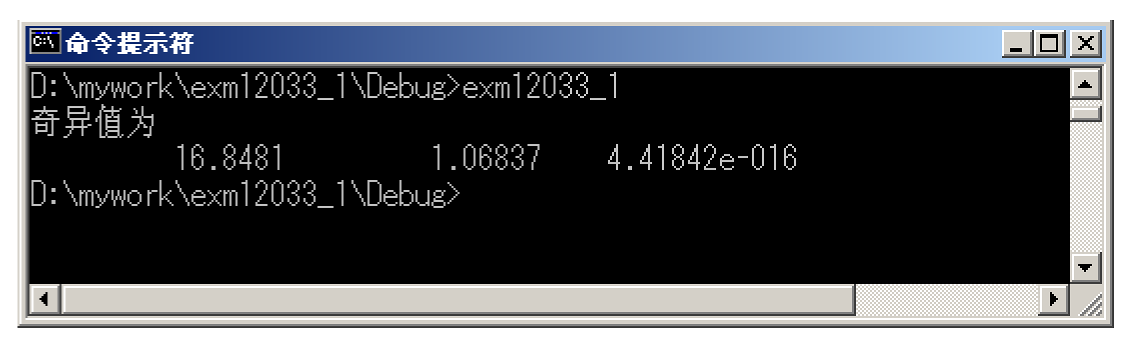


图12.3-9

12. 4 MATLAB 中 ActiveX 技术的应用

12.4.1 关于 ActiveX 的一般性说明

12.4.2 MATLAB 的 ActiveX 自动化

12. 4. 2. 1 MATLAB ActiveX 自动化控制器

【例 12.4.2-1】MATLAB 以自动化客户的资格通过 M 脚本文件把 Microsoft Excel 用作自动化服务器。本例演示调用自动化服务器的基本指令：Excel 缺省界面的开启；增添工作簿（Workbook）；改变激活的当前页（Worksheet）；MATLAB 与 Excel 之间的数据传递；Excel 的数据保存。

```
% exm12042_1.m
excel = actxserver('Excel.Application');
                                    %
                                    %
disp('为看清 Excel 界面及其变化，请把 MATLAB 界面调整得远小于屏幕！')
disp('按任意键，将可见到“Excel 界面”出现。')
pause
set(excel, 'Visible', 1);
disp('按任意键，可见到 Excel 界面出现第一张表激活的“空白工作簿”。')
pause
wkbs = excel.Workbooks;
Wbk = invoke(wkbs, 'Add');
disp('按任意键，当前激活表由第一张变为指定的第二张。')
pause
Sh = excel.ActiveWorkBook.Sheets;
sh2 = get(Sh, 'Item', 2);
invoke(sh2, 'Activate');
disp('按任意键，把 MATLAB 空间中的 A 矩阵送到 Excel 的指定位置。')
pause
Actsh = excel.Activesheet;
A = [1,2 ;3,4];
actshrng = get(Actsh,'Range','A1','B2');
set(actshrng, 'Value', A);                                    %  <23>
disp('按任意键，获取 Excel 指定区域上的数据，')
disp('并以 MyExcel.xls 文件形式保存在 D:\mywork 目录上。')
pause
rg = get(Actsh, 'Range', 'A1', 'B2');
B = rg.value;
B = reshape([B{:}], size(B));
invoke(Wbk, 'SaveAs', 'D:\mywork\myfile.xls');

disp('按任意键，关闭 excel 句柄代表的 Excel。')
pause
```

invoke(excel, 'Quit');

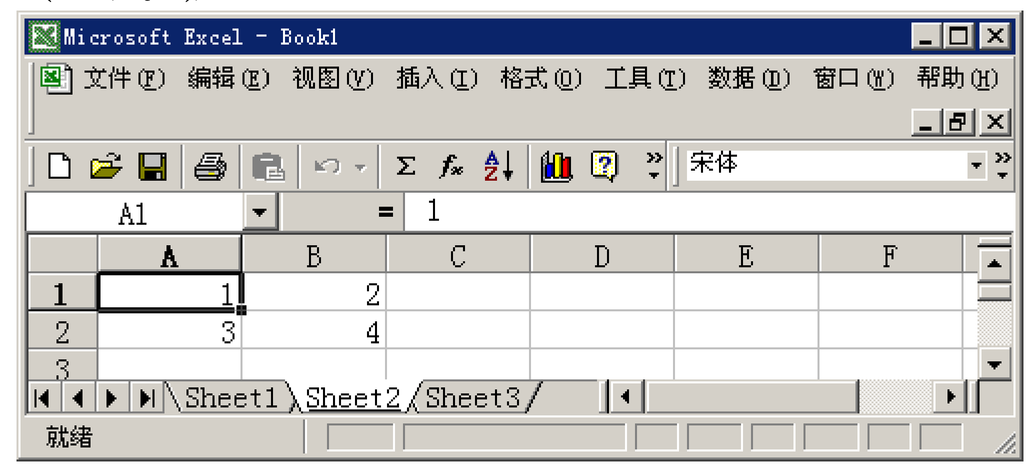


图 12.4-1

12.4.2.2 MATLAB ActiveX 自动化服务器

【例 12.4.2.2-1】创建一个名为 exm120422.ppt 的演示文稿。它包含一张如图 12.4-2 所示的幻灯片。该幻灯片在放映时，若在左上方空白框中输入任何 MATLAB 的合法指令，尔后按下计算按键，则在下方空白框中能实时地显示运算结果，如图 12.4-10。

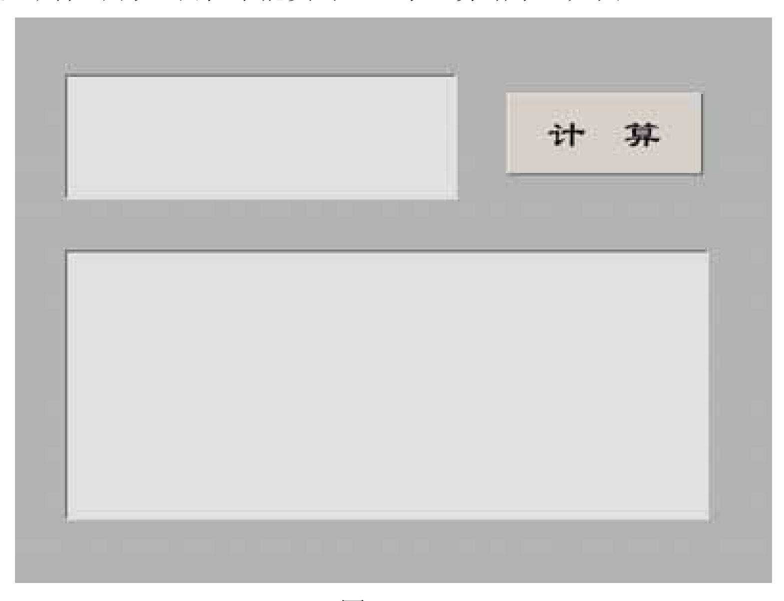


图 12.4-2

（1）

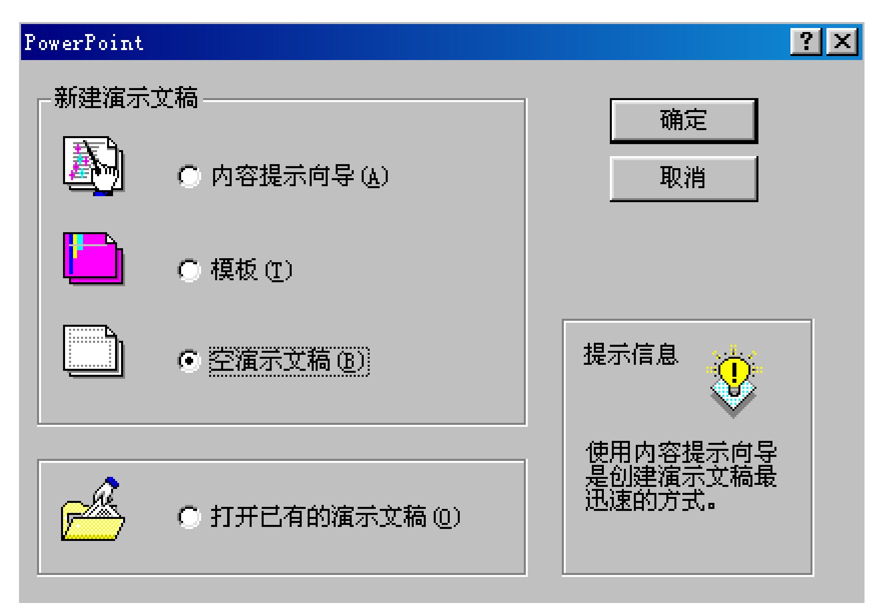

PowerPoint
新建演示文稿
内容提示向导(A)
模板(T)
空演示文稿(B)
打开已有的演示文稿(O)
确定
取消
提示信息
使用内容提示向导是创建演示文稿最迅速的方式。

图 12.4-3

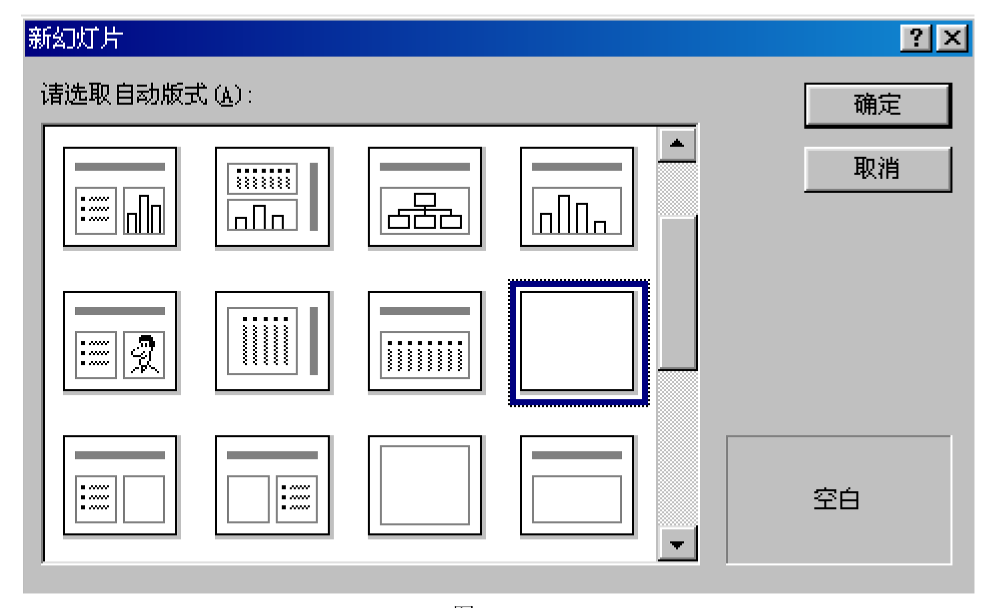

新幻灯片
请选取自动版式(A):
确定
取消
空白

图 12.4-4

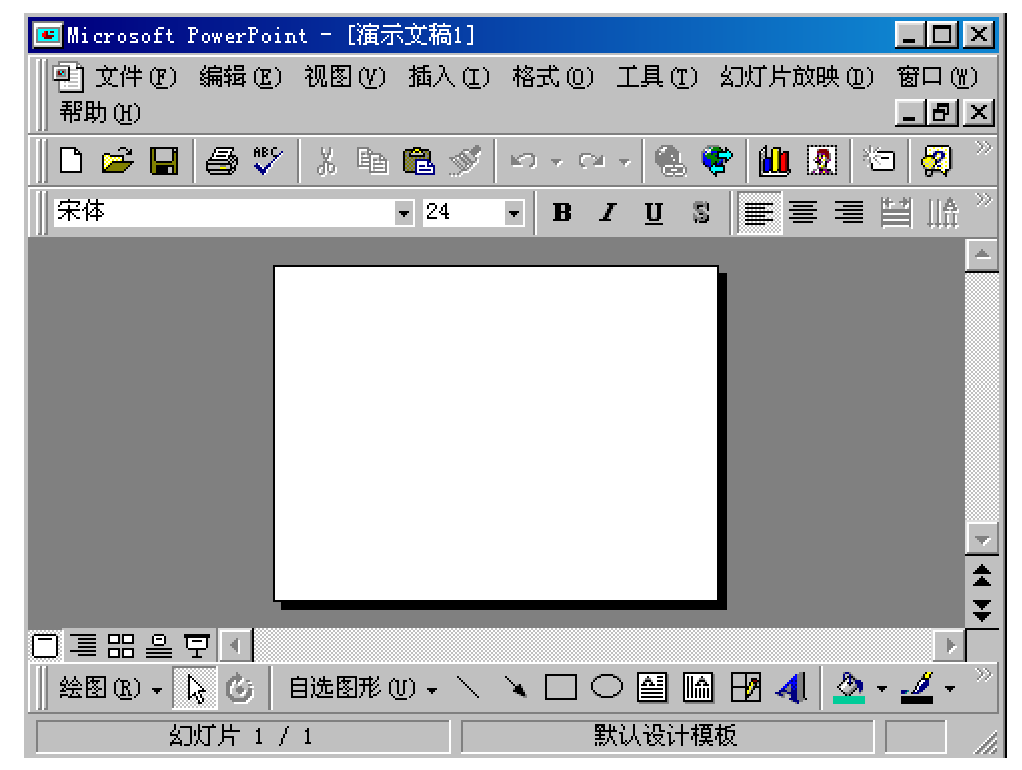


图 12.4-5

（2）

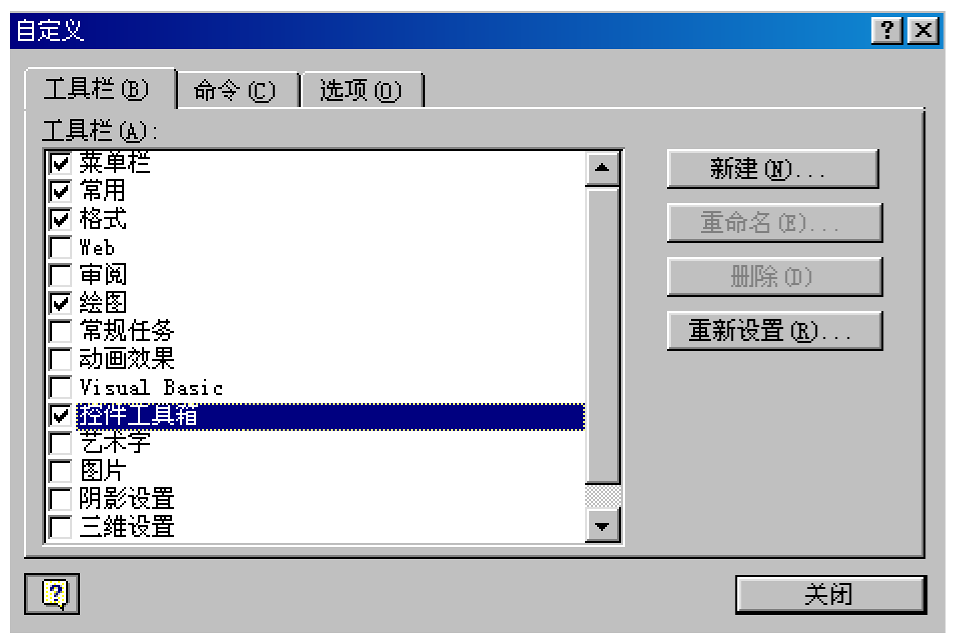


图 12.4-6

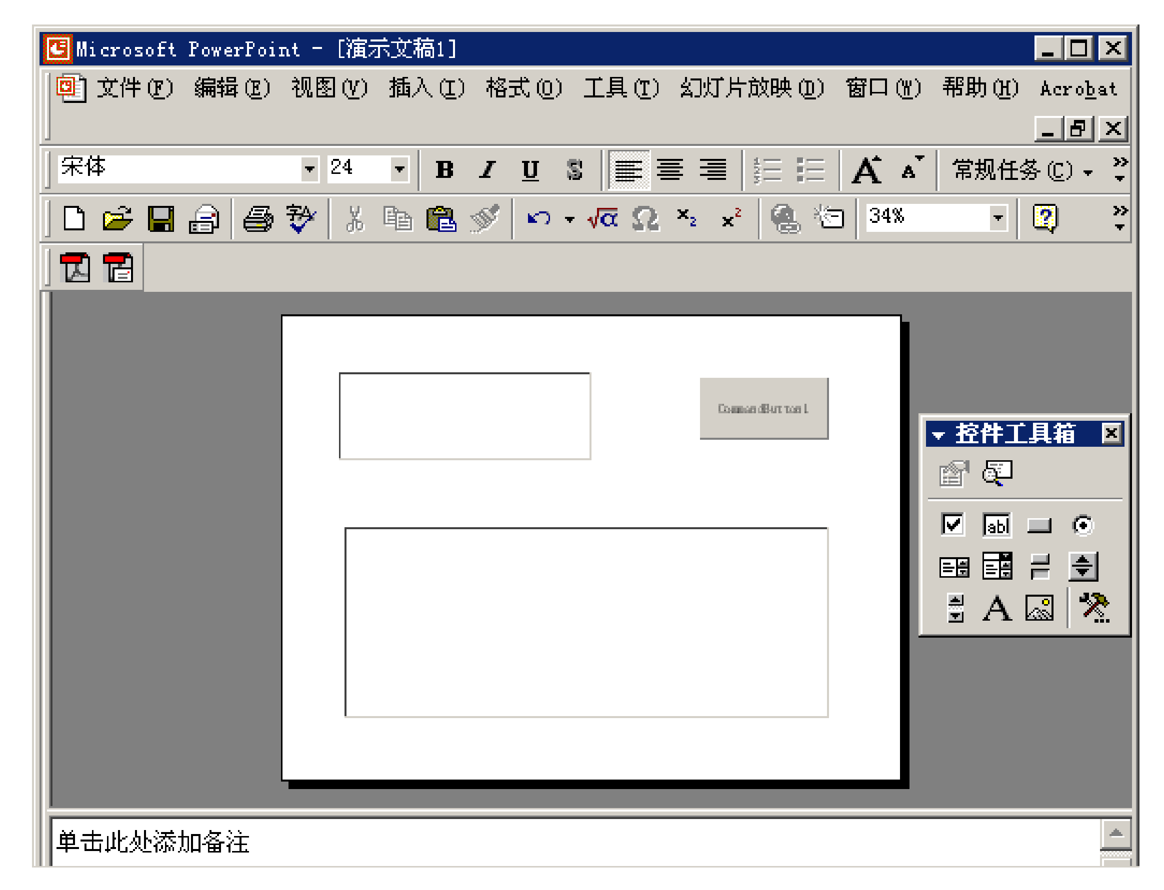


图 12.4-7

（3）

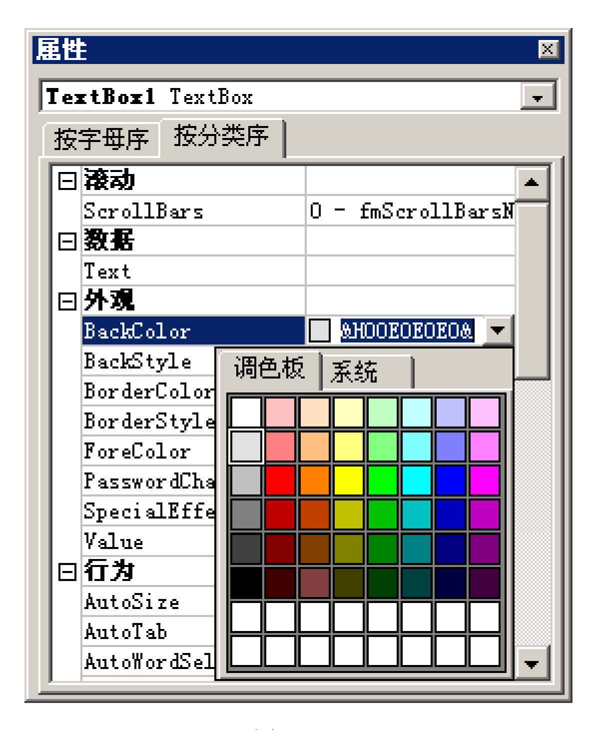


图 12.4-8

（4）

```
Private Sub CommandButton1_Click( )

End Sub
```

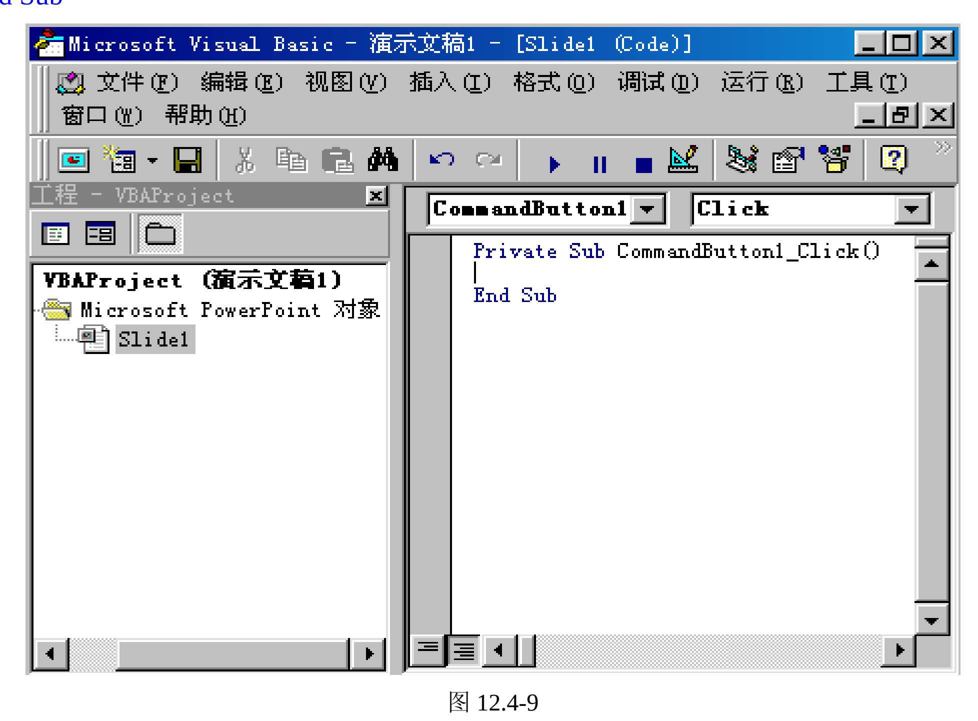


图 12.4-9

```
Rem
Dim h As String
Dim result As String
Rem
Dim matlab As Object
Set matlab = CreateObject("Matlab.Application")
Rem
h = TextBox1.Value
Rem
result = matlab.Execute(h)
Rem
TextBox2.Value = result
```

（5）

```
clc,rand('state',1)
A=rand(3,3),Cond_A=cond(A);
disp('The condition number of A')
disp(Cond_A)
```

计 算

```
A =
    0.9528    0.5982    0.8368
    0.7041    0.8407    0.5187
    0.9539    0.4428    0.0222
The condition number of A
    6.8261
```

图 12.4-10

【例 12.4.2.2-2】本例在上例创建的 exm120422.ppt 演示文稿中加入一张幻灯片，用于把 MATLAB 服务器绘置的图形嵌入到幻灯片放映窗口中。



图 12.4-11

（1）

（2）

（3）

Microsoft DirectAnimation Structured Graphics
Microsoft DirectAnimation Windowed Control
Microsoft Flat Scrollbar Control, version 6.0
Microsoft FlexGrid Control, version 6.0
Microsoft Forms 2.0 CheckBox
Microsoft Forms 2.0 ComboBox
Microsoft Forms 2.0 CommandButton
Microsoft Forms 2.0 Frame
Microsoft Forms 2.0 Image
Microsoft Forms 2.0 Label
Microsoft Forms 2.0 ListBox
Microsoft Forms 2.0 MultiPage
200 个控件

图 12.4-12

（4）

（5）

```
    Private Sub CommandButton1_Click( )

    End Sub


    Rem
Dim h As String
Dim result As String
    Rem
Dim matlab As Object
Set matlab = CreateObject("Matlab.Application")
    Rem
    Rem
result = matlab.Execute("set(gcf,'visible','off');")
h = TextBox1.Value
result = matlab.Execute(h)
result = matlab.Execute("print(gcf,'-dtiff','c:\a.tif');")
result = matlab.Execute("x=imread('c:\a.tif');")
result = matlab.Execute("imwrite(x,'c:\a.bmp');")
Image1.Picture = LoadPicture("c:\a.bmp")
SlideShowWindows(1).View.GotoSlide 2
result = matlab.Execute("delete c:\a.tif")
result = matlab.Execute("delete c:\a.bmp")
```

（5

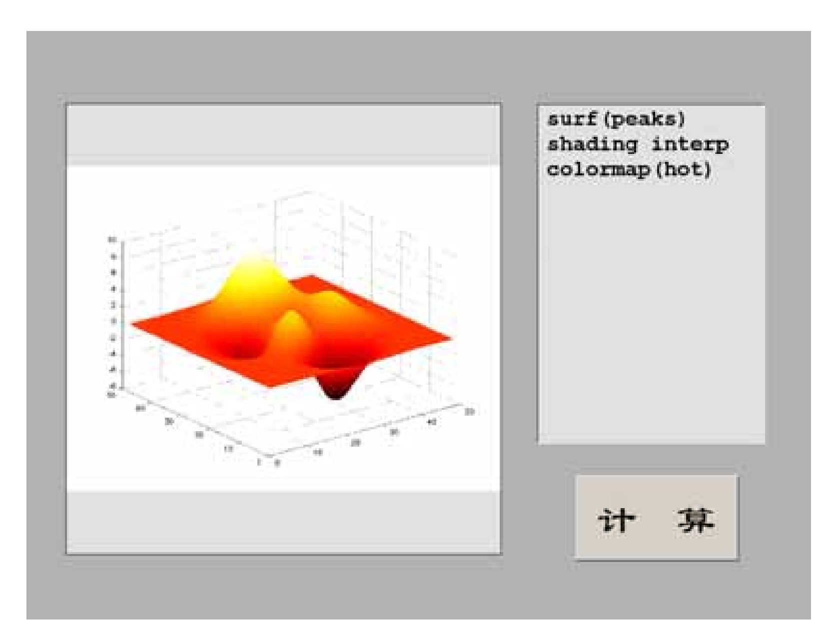


图 12.4-13

12.5 MATLAB 中 DDE 技术的应用

12.5.1 关于 DDE 的一般性说明

12.5.2 DDE 中的 MATLAB 服务器

【例 12.5.2-1】利用 Visual Basic 制作如图 12.5-3 所示的图形用户界面，在该界面中，可以输入任何 MATLAB 指令，点击【计算】按键后可输出相应计算结果。该界面运行时，把 MATLAB 作为 DDE 的服务器使用。

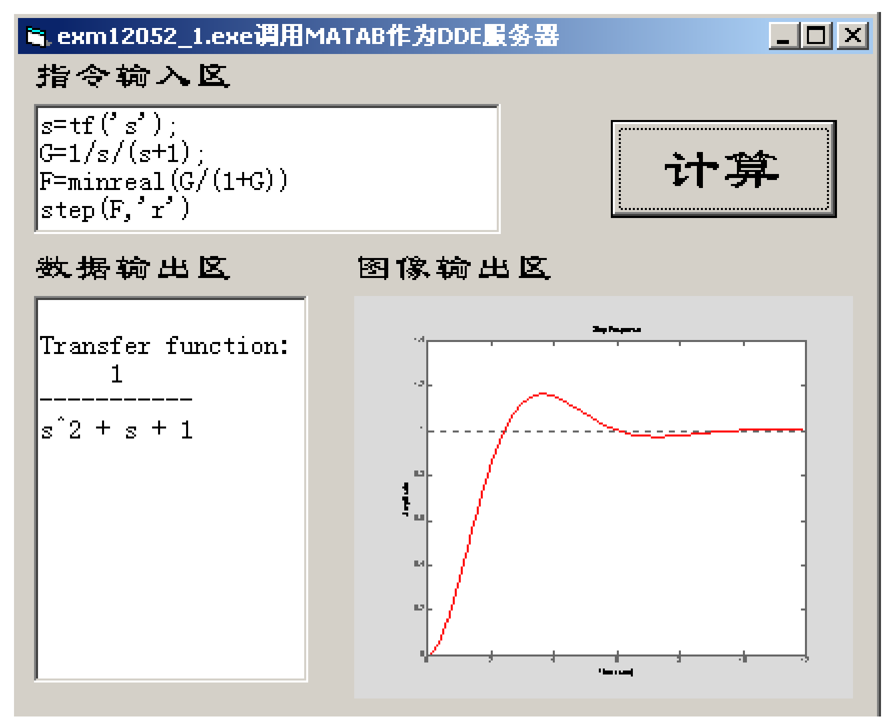


图 12.5-3

（1）

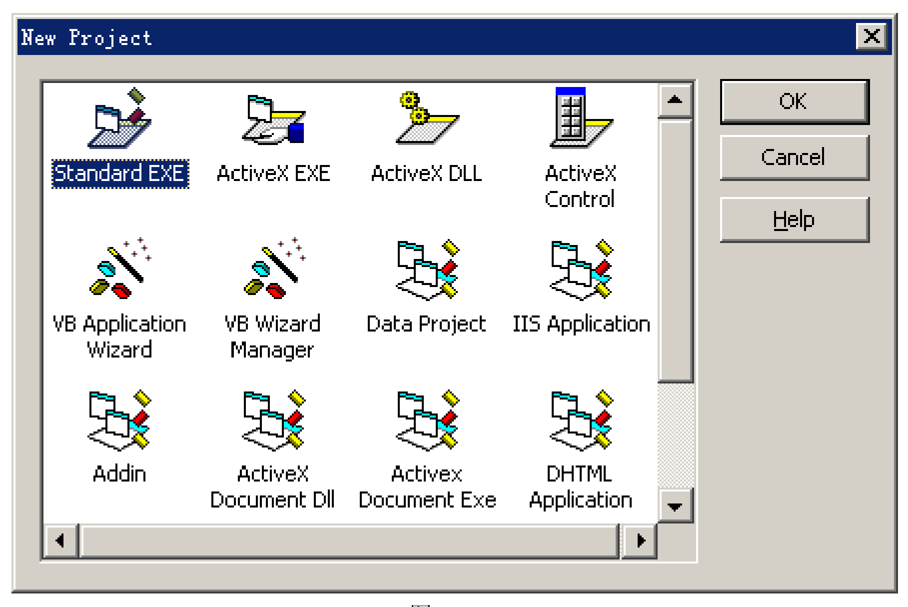


图 12.5-4

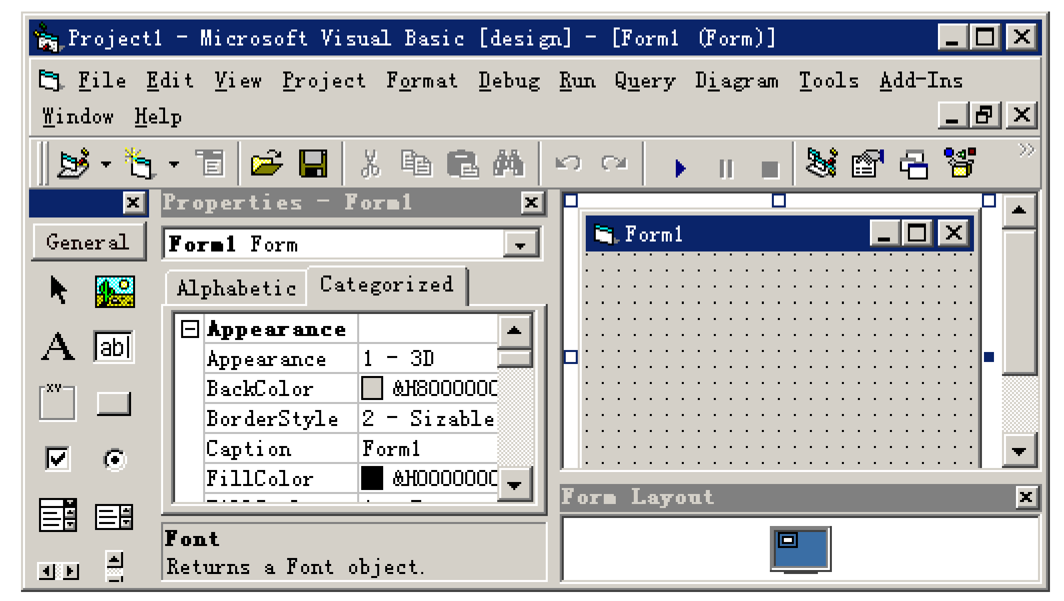


图 12.5-5

（2）

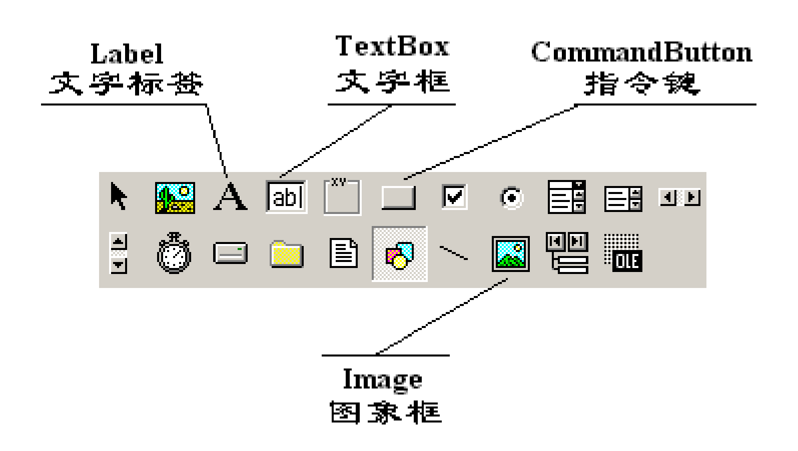


图 12.5-6

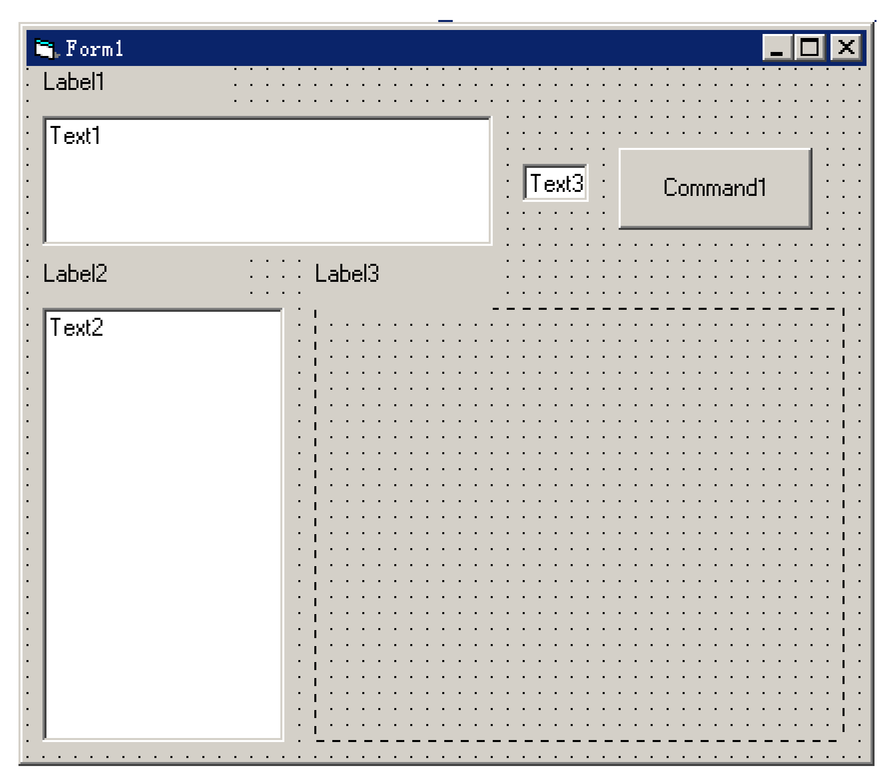


图 12.5-7

（3）

（4）

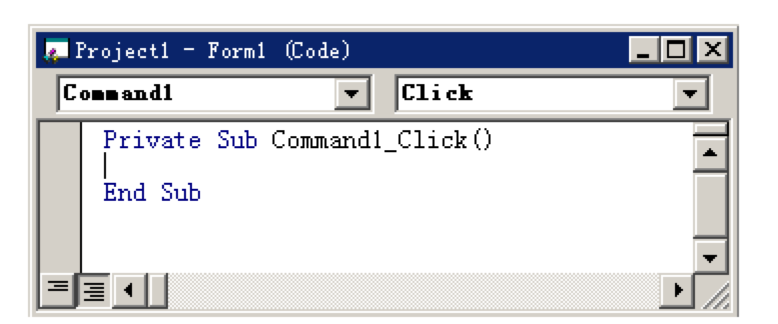


图 12.5-8

```
TextInput.LinkMode = vbLinkNone
TextInput.LinkTopic = "MATLAB|Engine"
TextInput.LinkItem = "EngEvalString"
TextInput.LinkMode = vbLinkManual

szCommand = TextInput.Text
TextInput.LinkExecute szCommand
TextInput.LinkMode = vbLinkNone

FigText.LinkMode = vbLinkNone
FigText.LinkTopic = "MATLAB|Engine"
```

```
FigText.LinkItem = "EngFigureResult"
FigText.LinkMode = vbLinkManual

FigText.LinkRequest
If (FigText.Text = "yes") Then
    Image1.Picture = Clipboard.GetData()
    Else
    Image1.Picture = LoadPicture
End If

TextOutput.LinkMode = vbLinkNone
TextOutput.LinkTopic = "MATLAB|Engine"
TextOutput.LinkItem = "EngStringResult"
TextOutput.LinkMode = vbLinkManual
TextOutput.LinkRequest
TextOutput.LinkMode = vbLinkNone
```

（5）

12.5.3 DDE 中的 MATLAB 客户

【例 12.5.3-1】设计一个 DDE 对话程序。MATLAB 作为客户，Excel 作为服务器。本例演示：（A）DDE 的创建和关闭；（B）热连接的建立和使用。

（1）

```
% exm12053_1.m
clear
h=surf(peaks(20));
z=get(h,'zdata');
chann=ddeinit('excel','Sheet3');
range2='r1c1:r20c20';
rc=ddepoke(chann,range2,z);
%.
rc=ddeadv(chann,range2,'set(h,''zdata'',z);','z');
%
hc=uimenu(gcf,'Label','关闭');
hc1=uimenu(hc,'Label','关热连接','Callback','ddeunadv(chann,range2);');
hc2=uimenu(hc,'Label','关闭对话','Callback','ddeterm(chann);');
hc3=uimenu(hc,'Label','关图形窗','Callback','close;');
```

（2）

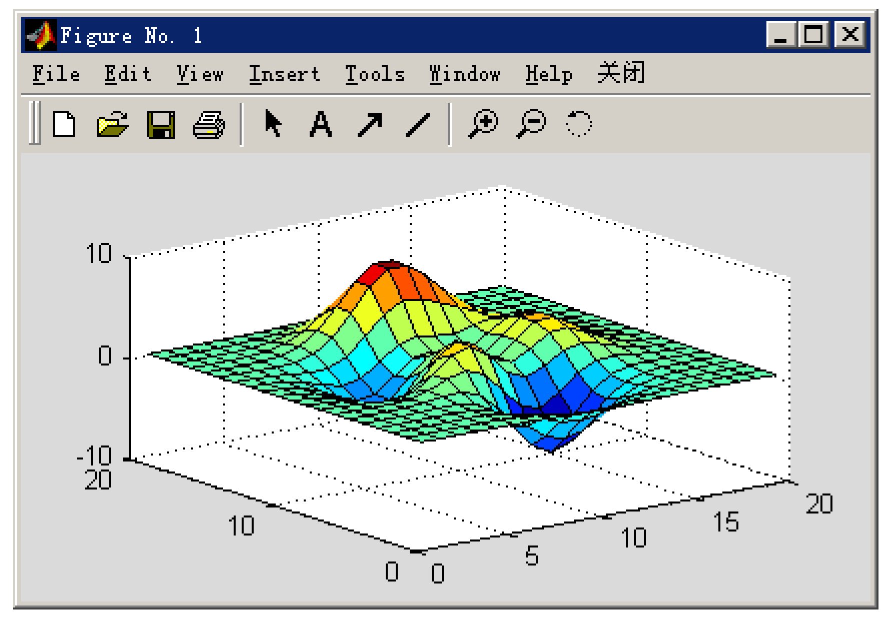
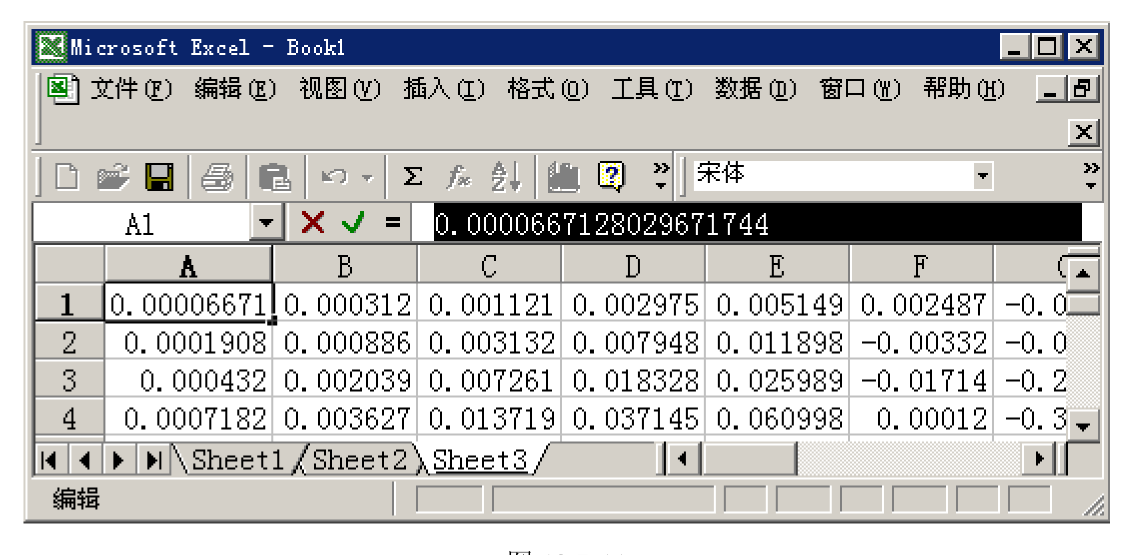


图 12.5-10



	A	B	C	D	E	F	G
1	0.00006671	0.000312	0.001121	0.002975	0.005149	0.002487	-0.0
2	0.0001908	0.000886	0.003132	0.007948	0.011898	-0.00332	-0.0
3	0.000432	0.002039	0.007261	0.018328	0.025989	-0.01714	-0.2
4	0.0007182	0.003627	0.013719	0.037145	0.060998	0.00012	-0.3

图 12.5-11

（3）

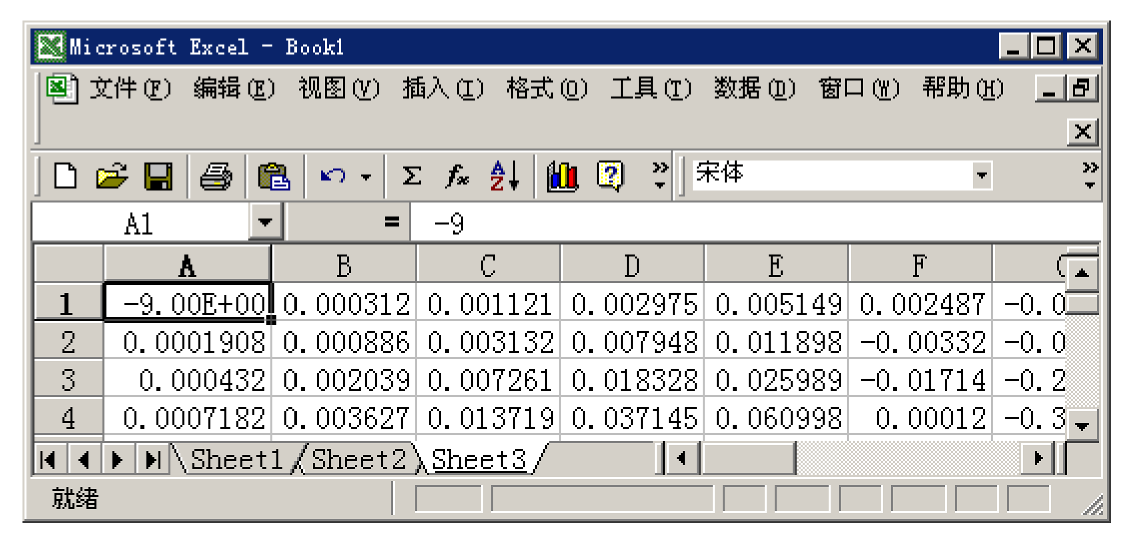


图 12.5-12

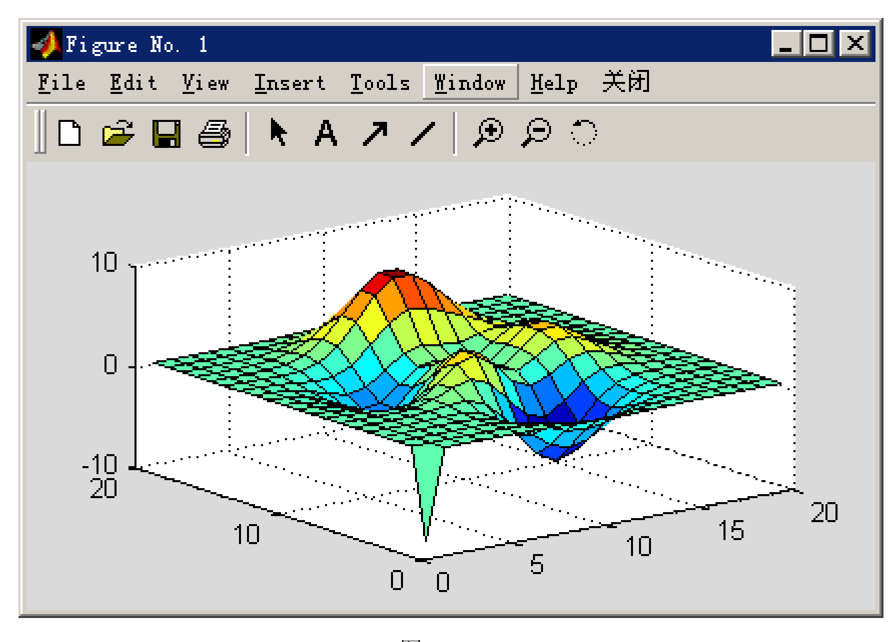


图 12.5-13

（4）

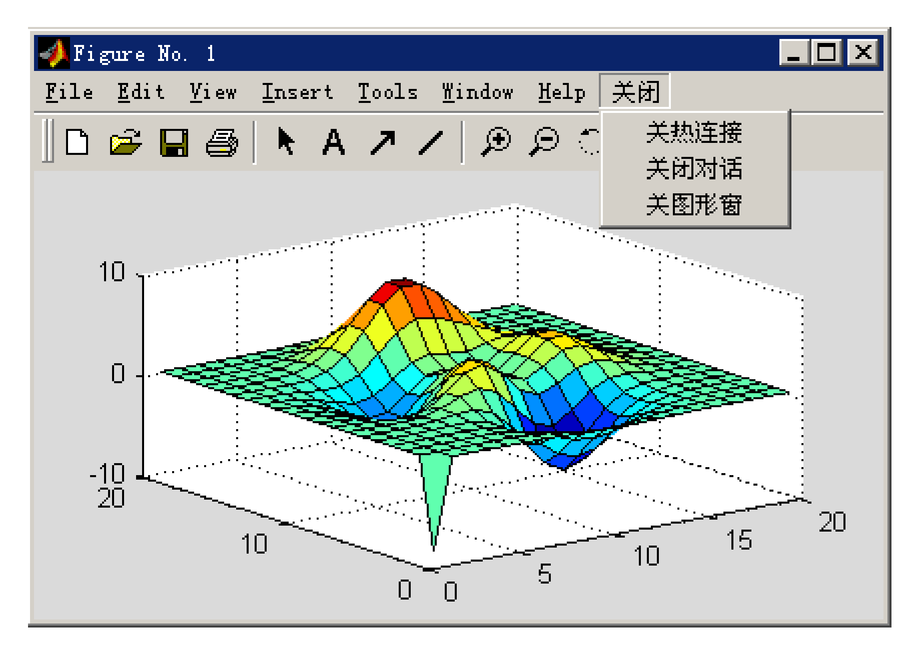

Figure No. 1
File
Edit
View
Insert
Tools
Window
Help
关闭
关热连接
关闭对话
关图形窗
10
0
-10
20
10
0
0
5
10
15
20

图 12.5-14

第十三章 Notebook

Notebook 的功能在于：使用户能在 Word 环境中“随心所欲地享用”MATLAB 的浩瀚科技资源，为用户营造融文字处理、科学计算、工程设计于一体的完美工作环境。

MATLAB Notebook 制作的 M-book 文档不仅拥有 MS-Word 的全部文字处理功能，而且具备 MATLAB 无与伦比的数学解算能力和灵活自如的计算结果可视化能力。它既可以看作解决各种计算问题的字处理软件，也可以看作具备完善文字编辑功能的科技应用软件。

M-book 文档最显著的特点是它的“活”性：

- 在科技报告、论文、著作和讲义教材的撰写过程中，为作者营造了文字语言思维和科学计算思维的和谐氛围。
- 用 M-book 写成的电子著作、电子讲义、网上教材不仅图文并茂，而且动静结合。那些由 MATLAB 指令构成的例题、演示，都可供读者亲自操作，举一反三，从而在“手脑并用”的环境中由此及彼、由浅入深。
- M-book 文档能“无缝”地与 PowerPointer 、Authorware 等应用软件相链，使计算机演讲不仅使听讲者看到事先编排的“幻灯片”和“影片”，而且可以让听讲者看到实时科学计算结果，增加听讲者的临场感、参与感。

本章叙述由 MATLAB 6.x 和 MS Word 组合成的中文 Notebook 环境，兼顾其他版本。在保证内容完整的前提下，本章围绕 Notebook 使用中的要点、难点展开，并强调可操作性。一般性资料，请读者查阅 MATLAB 随带文件 matlab\help\pdf_doc\matlab\notebook.pdf。

13.1 Notebook 的安装

13.1.1 MATLAB 6.1 版 Notebook 的安装

13.1.2 MATLAB 6.5 版 Notebook 的安装

13.2 Notebook 的启动

13.2.1 从 Word 中启动 Notebook

13.2.2 从 MATLAB 中启动 Notebook

13.2.3 Notebook 成功启动标志和中文 M-book 模板的形成

13.2.3.1 Notebook 成功启动的直观标志

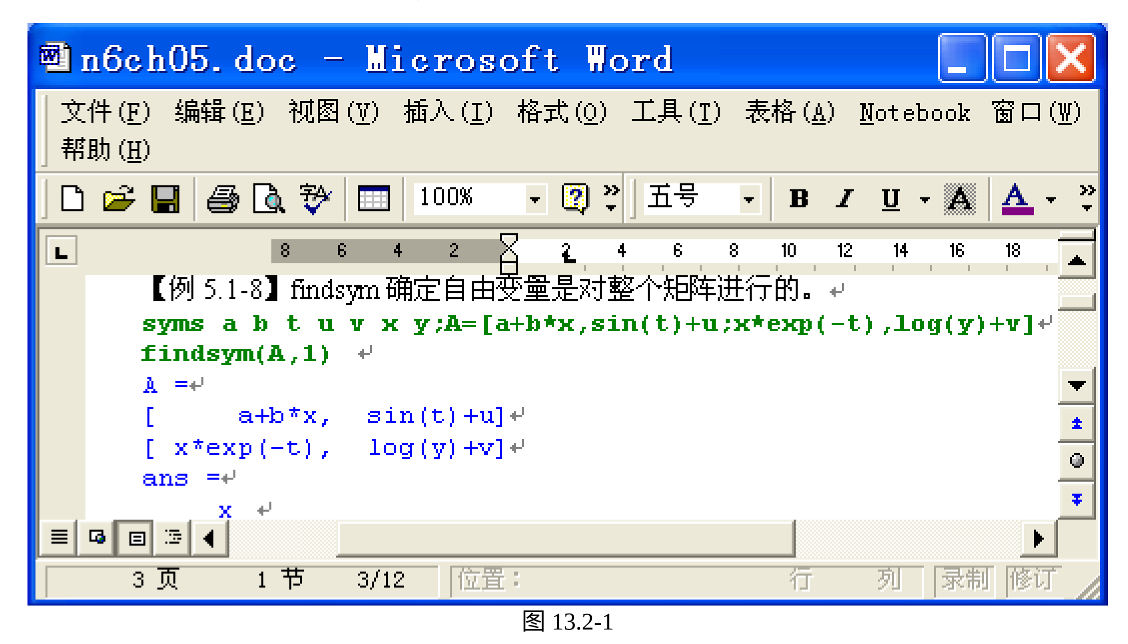


图 13.2-1

13.2.3.2 中文 M-book 模板的初始化

（1）**Notebook 初次使用时的宏安全设置**
（2）**M-book 在中文环境中的初貌**
（3）**中文 M-book 模板的生成步骤**

13.2.4 启动失败的原因和解决办法

13.2.4.1 Notebook 启动失败的原因

13.2.4.2 本书提供的启动解救文件

```
function mbookzzy(filename)
% mbookzzy.m
% filename
%
%

wp1='progra~1';
wp2='micros~2';
wtm='templa~1';
wpathzzy=['c:\',wp1,'\',wp2,'\office\winword.exe '];
templatezzy=['c:\',wp1,'\',wp2,'\',wtm,'\m-book.dot '];
```

```
czzy='/mmwNewNotebookFromCmdLines&';
switch(nargin)
case(0)
   dos([wpathzzy,templatezzy,czzy]);
case(1)
   if ~exist(filename)
      error('指定文件不在路径上或不存在 。')
   end
   dos([wpathzzy,filename,'&']);
end
```

13.3 M-book 模板的使用

13.3.1 输入细胞（群）的创建和运行

13.3.1.1 细胞（群）

13.3.1.2 基本操作

13.3.1.3 输入细胞（群）操作示例

【例 13.3.1.3-1】演示：创建并运行输入细胞的基本操作方法。

（1）

```
xx=(1:5)/5*pi;yy=sin(xx).*exp(xx)
```

（2）

```
x=(1:4)/4*pi;y=sin(x).*exp(x)
y =
    1.5509    4.8105    7.4605    0.0000
```

【例 13.3.1.3-2】演示：文本中内嵌输入细胞的操作方法。

运行符号计算指令：`syms x y; f=x^3*y+y^0.5; Dfdxdy=diff(diff(f,x),y), S=int(int(f,y,1,x^2),x,1,2)` 运行后可得到导数 $\frac{df(x,y)}{dxdy}$ 和二重积分 $S=\int_1^2\int_1^{x^2}(x^3y+\sqrt{y})dydx$ 的准确结果

```
Dfdxdy =
3*x^2
S =
763/48
```

【例 13.3.1.3-3】演示：生成完整图形的多条图形指令必须定义在同一细胞（群）中。

```
t=0:0.1:10;y=1-cos(t).*exp(-t);                              %<1>
tt=[0,10,10,0];
yy=[0.95,0.95,1.05,1.05];
fill(tt,yy,'g'),axis([0,10,0,1.2]),xlabel('t'),ylabel('y')     %<4>
hold on                                                      %<5>
```

```
plot(t,y,'k','LineWidth',4)                                   %<6>
hold off                                                      %<7>
ymax=max(y)                                                   %<8>
ymax =
    1.0669
```

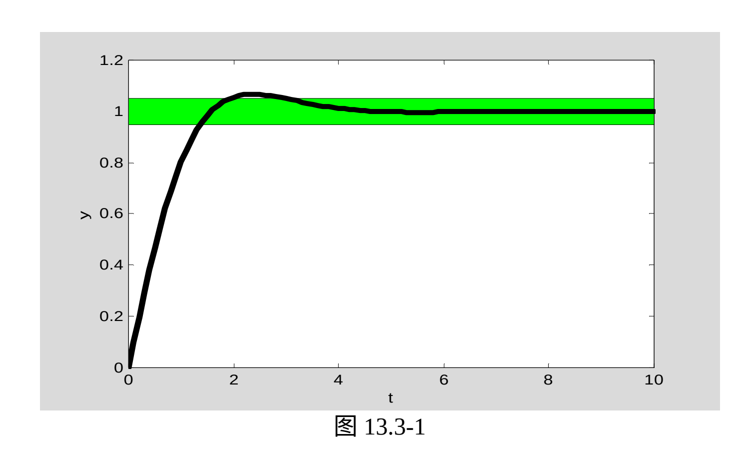


图 13.3-1

13.3.2 计算区的创建和运行

13.3.2.1 计算区和使用要旨

13.3.2.2 计算区的基本操作

13.3.2.3 形成计算区的算例

[ ]

【例 13.3.2.3-1】 本例专门为演示计算区而设计。

```
G1=tf([1 0.5],[1 4]);
G2=tf(1,[500 0 0]);
H=tf([1 1],[1 2]);
S=tf(minreal(G1*G2/(1+G1*G2*H)))                              %<4>

Transfer function:
       0.002 s^2 + 0.005 s + 0.002
-----------------------------------------
s^4 + 6 s^3 + 8.002 s^2 + 0.003 s + 0.001
```

13.3.3 Notebook 菜单的其他选项

13.3.3.1 自初始化细胞及其应用

13.3.3.2 细胞的循环运行

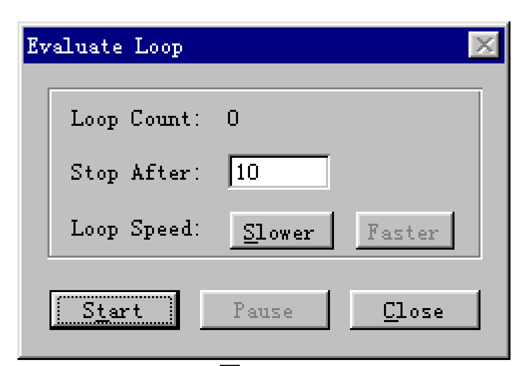


图 13.3-3

【例 13.3.3.2-1】利用【Evaluate Loop】菜单选项绘制图形。

（1）

```
clear;x=0:10;k=1;hold on
```

（2）

```
y=k*x;
plot(x,y)
k=k+1;
```

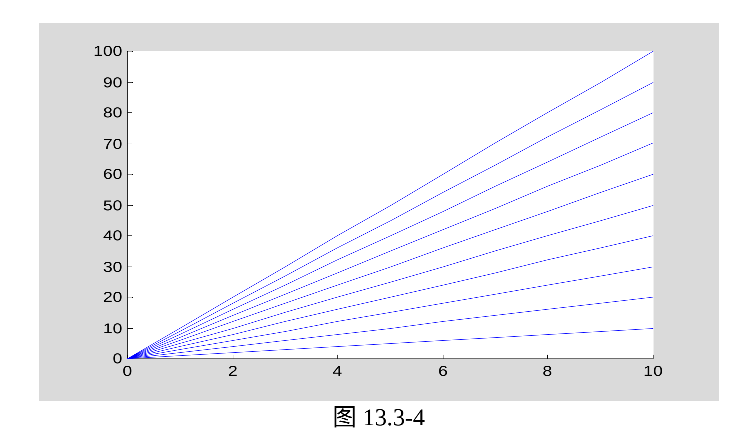


图 13.3-4

（3）

13.3.3.3 整个 M-book 文件的运行

13.3.3.4 删去 M-book 文件所有输出细胞

13.3.4 输出细胞的格式控制

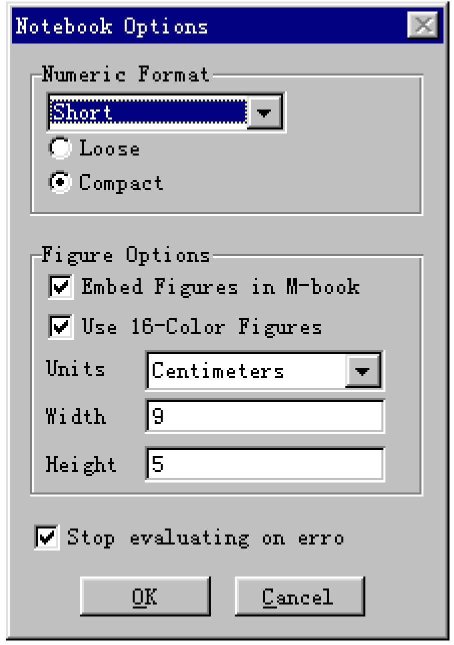


图 13.3-5

13.3.4.1 输出数据的表示法

13.3.4.2 输出数据间的空行控制

13.3.4.3 图形的嵌入控制

【例 13.3.4.3-1】演示：两种图形控制作用的操作和不同影响。

（1）

```
surf(peaks);colormap(hot)
t=(0:50)/50*pi;y=sin(t);plot(t,y)
```

（2）

```
surf(peaks);colormap(hot)
t=(0:50)/50*pi;y=sin(t);plot(t,y)
```

（3）

```
surf(peaks);colormap(hot)    (no graph)
t=(0:50)/50*pi;y=sin(t);plot(t,y)
```

13.3.4.4 嵌入图形大小的控制

13.3.4.5 嵌入图形的背景色问题

13.3.4.6 嵌入图形的打印输出问题

13.3.4.7 M-book 处理活动画面的能力

【例 13.3.4.7-1】在 M-book 中尝试运行以下几组指令。（这些指令都是前面章节给出的动画函数文件名。）

（1）

```
fly_zzy
```

（2）

```
fly_zzy2
```

（3）

```
whitebg('white');anim_zzy1(1)
```

（4）

```
shg;textzzy('ABC')
```

13.3.5 细胞的样式

13.3.6 使用 M-book 模板的若干参考技法

13.4 科技演讲稿的制作

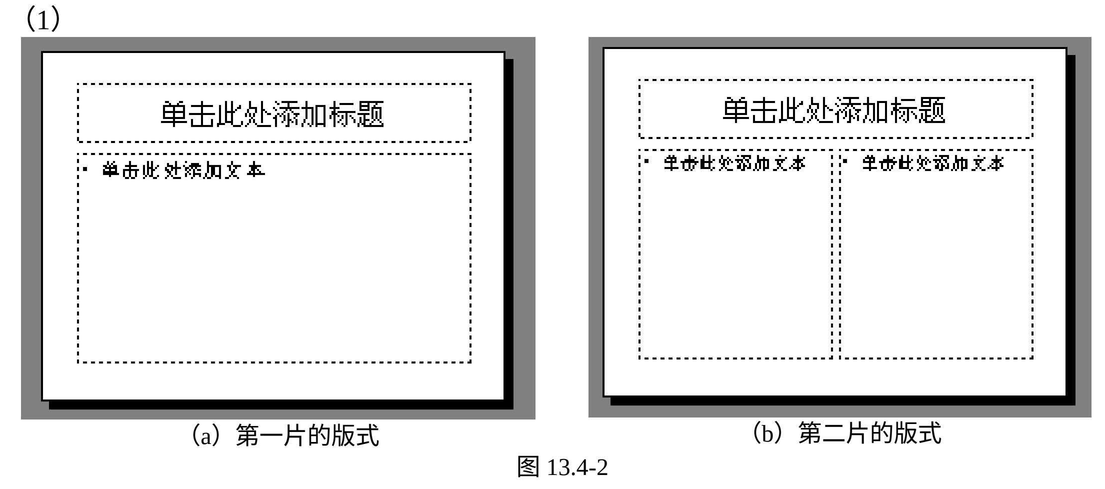
13.4.1 在 M-book 中实现计算和可视的演讲稿制作



图 13.4-1

【例 13.4.1-1】制作可引导本书光盘上 doc 文件的 PowerPoint 幻灯片文件 **E_book.ppt**。

(1)



(a) 第一片的版式　　(b) 第二片的版式

图 13.4-2

13.4.2 直接引出 GUI 图形用户界面的演讲稿制作

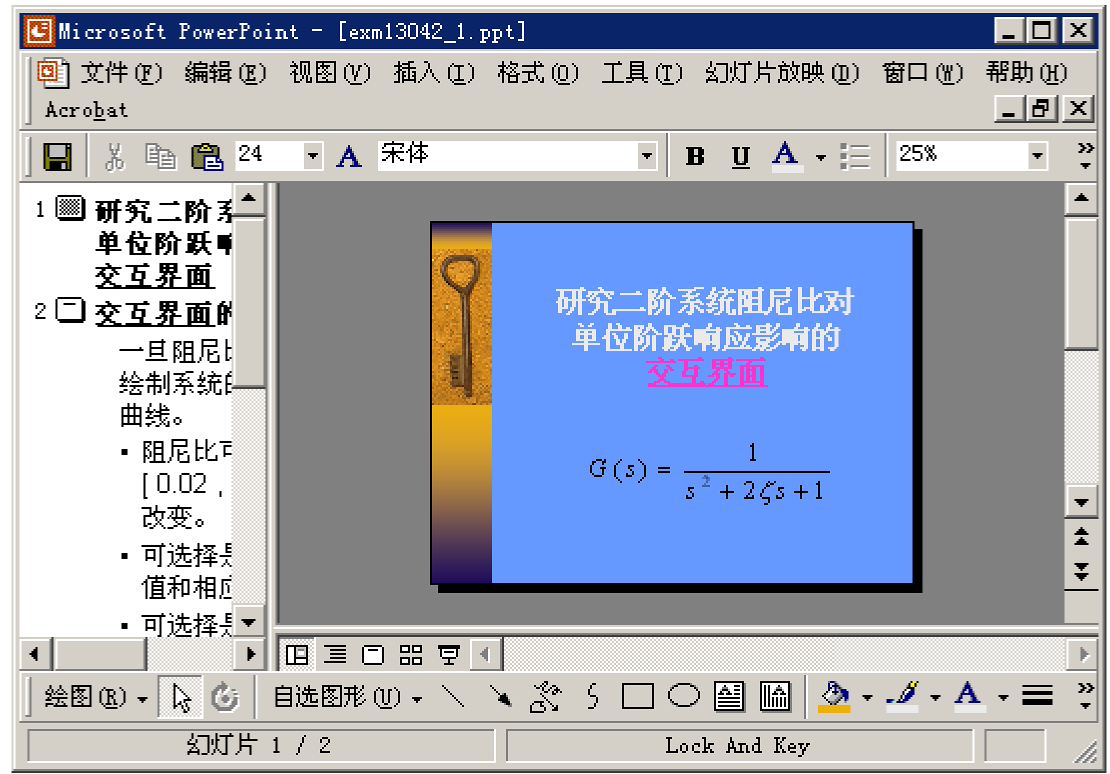


图 13.4-3

【例 13.4.2-1】制作能直接引出 MATLAB 图形用户界面的 PowerPoint 演讲稿 **exm13042_1.ppt**。

13.4.3 具有现场计算和绘图能力的演讲文稿的制作

【例 13.4.3-1】制作演示 Laplace 变换的演讲稿。要求：在普通的文字幻灯片中嵌入“能实时进行 Laplace 运算的幻灯片”。

（1）

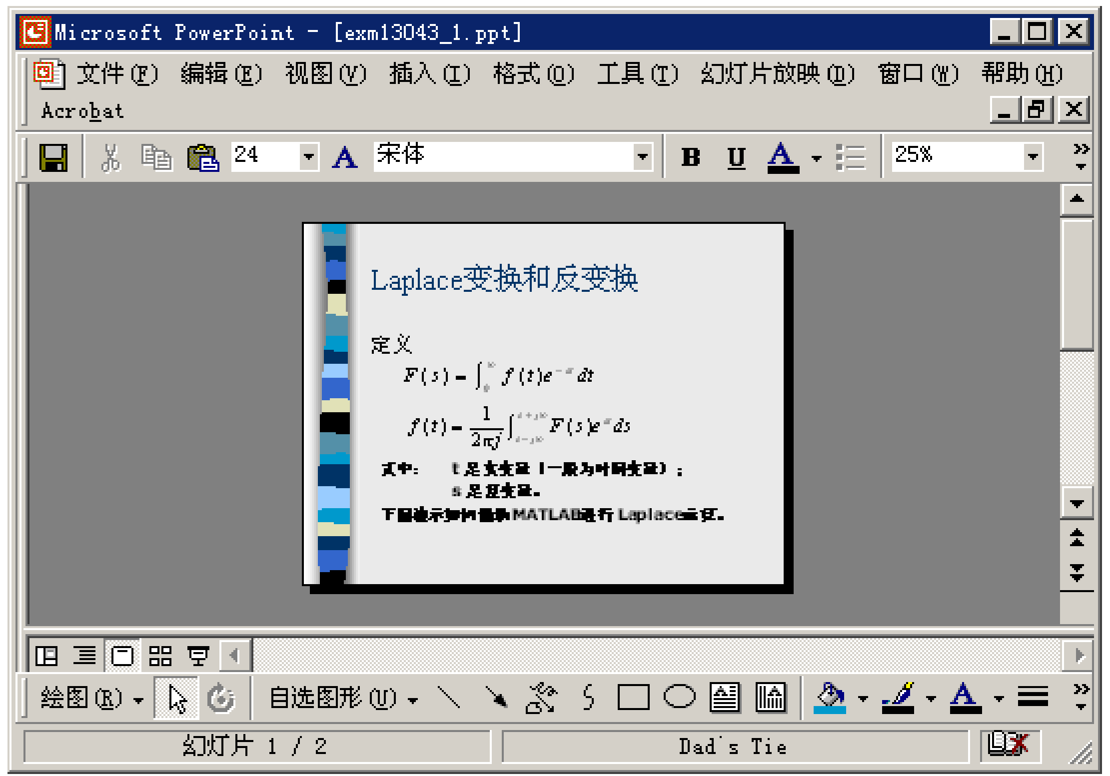


图 13.4-4

(2)

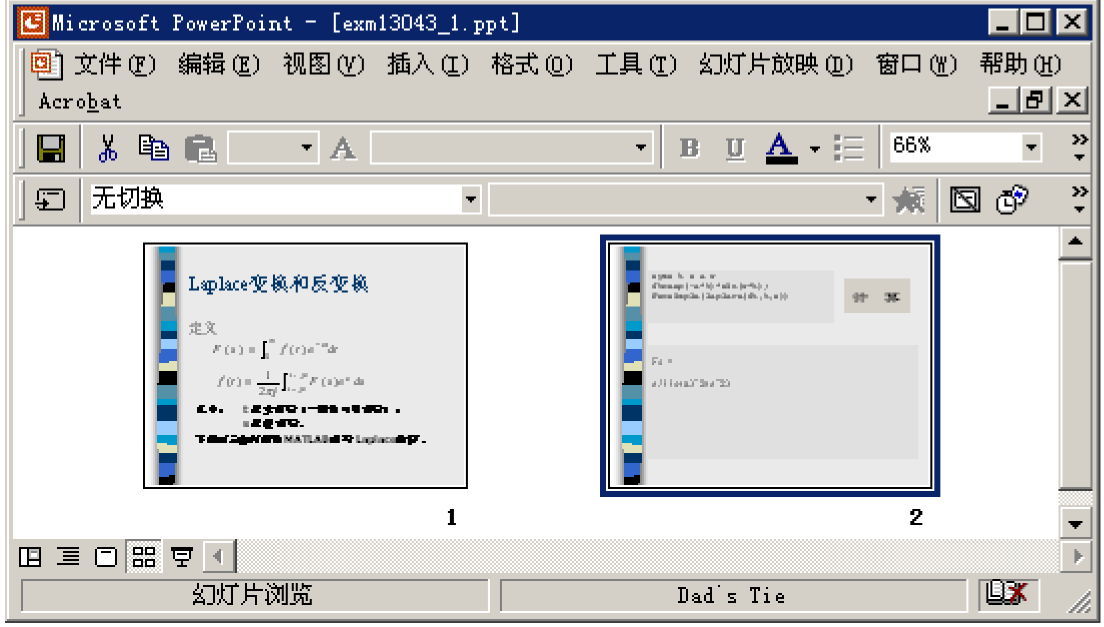


图 13.4-5

(3)

```
syms t s a w
ft=exp(-a*t)*sin(w*t);
Fs=simple(laplace(ft,t,s))
```

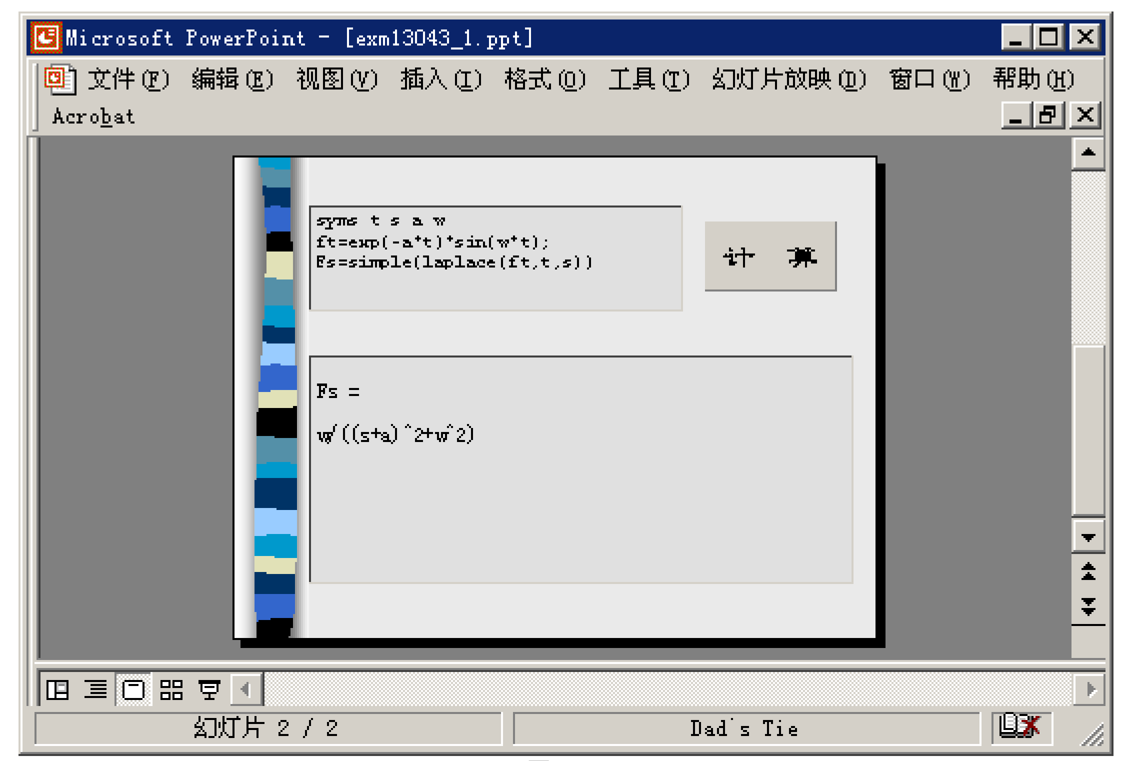

Microsoft PowerPoint - [exm13043_1.ppt]
文件(F) 编辑(E) 视图(V) 插入(I) 格式(O) 工具(T) 幻灯片放映(D) 窗口(W) 帮助(H)
Acrobat
syms t s a w
ft=exp(-a*t)*sin(w*t);
Fs=simple(laplace(ft,t,s))
计 算
Fs =
w/((s+a)^2+w^2)
幻灯片 2 / 2
Dad's Tie

图 13.4-6

附录 B：　光盘使用说明

B.1 光盘文件的结构

在光盘上有如下五个文件夹：

- **matlab_api**　　专放第 12 章所涉及的各类文件。
- **matlab_c**　　专放第 11 章所涉及的各类 M、C、DLL、EXE 文件。
- **mbook**　　存放着包含本书全部算例的 M-book 形式 DOC 文档。
- **mfiles**　　汇集了全书 90%以上算例的 M 或 MDL 文件。
- **PowerMatlab**　　存放着 PowerPoint 制作的科技演讲稿简例文件。

B.2 光盘对软件环境的要求

- 需要（包含 Word、PowerPoint 的）**Offiece2000** 和 **MATLAB6.5** 支持。
- 假如要运行 mbook 文件夹上的 DOC 文件，则需要 **Notebook** 环境。（关于 Notebook 环境的设置参见第 13.1 节）
- 假如要编译产生 EXE 文件，则需要与 **MATLAB6.5** 适配的 Borland 、MicroSoft 、Visual 、Watcom **C/C++**（详见 11.2 和 11.3 节）。
- 假若要制作 API 接口，则需要相应的配套软件。

B.3 光盘文件的操作准备

在运行光盘文件之前，应首先使用 MATLAB 的路径浏览器把 \mfiles，\matlab_c，\matlab_api 等三个文件夹设置在 MATLAB 的搜索路径上。这种设置可以是“永久”的，即今后再打开 MATLAB 时，总确认 \mfiles 文件夹等在搜索路径上。

假如在今后运行 MATLAB 时，光盘没有插入，那么在 MATLAB 指令窗中会出现警告性提示。这对光盘以外的 MATLAB 其他运作没有任何影响。

B.4 mbook 文件夹上 DOC 文件的使用

该文件夹包含第 1 章到第 13 章的全部算例，以及前言、附录等。章节的编号、名称与印刷版完全一致。

光盘 DOC 文件都是在 MATLAB 6.5 的 Notebook 环境中生成的。

（1）光盘 DOC 文件的功用

- **弥补了印刷版丢失的色彩信息**

  在 MATLAB 中，用 M 文件编辑器或 Notebook 编写的指令或文件运行的结果（尤其图形）都采用不同的色彩鲜明地表现对象特征。但目前印刷版书籍出于价格和技术原因，不得不牺牲色彩信息而采用“黑白”处理。读者借助本光盘可克服印刷版丢失色彩信息的遗憾。

- **提供了与印刷版对应的 Notebook 演练环境**

  本光盘中 DOC 文件的章节结构、算例编号与印刷版完全相同。因此在学习过程中，读者可在本光盘启动的 Notebook 环境中，或直接运行算例，观察运行结果；或改变若干指令，举一反三地观察运行结果的变化；或通过简单的复制操作，使相应指令在 MATLAB 指令窗中运行，而避免自己键入的错误。

- **提供制作 M-book 的样板**

读者若想制作自己的 M-book 文档，可调用本光盘任何一章的 DOC 文件为样板进行。注意：假若调用原英文 M-book 模版，则需通过若干设置才能在中文状态下正常运行。

（2）DOC 文件的开启

- 所有 DOC 文档都是在“**Word2000 中文版 + MATLAB6.1**”构成的 Notebook 环境中生成的。因此，在相同环境下开启是最佳选择。此时，文档具有“活性”。
- 假如读者的 MATLAB 与 Word 联接正确，用鼠标双击光盘上的 DOC 文件，就能直接进入 Notebook 环境。
- 若前述启动失败，可以先启动 MATLAB，然后再启动光盘 DOC 文档。在很多情况下，这样的操作很可能奏效。
- 在不具备相同 Notebook 环境的情况下，DOC 文档最好在 Word2000 环境中开启。此时电子文档虽然不“活”，但所有指令都能被准确地复制。
- 在 DOC 文件开启后，节次编号有可能显现得不很正确。这是由 Word 软件本身问题引起的。此时只要进行以下操作就可使编号正确：先使光标位于“章名”大标题上；然后点中工具条 标题 1 样式栏右边的下拉菜单按键，引出下拉菜单；在这下拉菜单中再点选“标题 1”菜单项，就能自动整理节次编号，使之正确。

（3）光盘 DOC 文件的使用方法

- **作为演练环境使用**

  在正常打开的光盘 DOC 文件中，读者只要把光标放在绿色的输入细胞内，按组合键【Ctrl + Enter】，就可使该输入细胞重新执行计算。在演练中，读者可以通过对指令的修改、变化和重新运行，观察运算结果的变化，从而达到举一反三的效果。
- **作为样板使用**

  先打开光盘 DOC 文件，然后删去原光盘文件内容，再写入读者自己所需的内容，最后通过菜单项的“另存为”操作保存为自己的文件。这样获得的文件能正常地在 Notebook 环境下工作，也就是既可以输入文字、公式，又可以运行 MATLAB 指令、嵌入数值或图形结果；既拥有 Word 的所有文字处理能力，又具备 MATLAB 的运算、表现能力。

B.5 mfiles 文件夹上的 M、MDL 文件的使用

除不能用 M、MDL 文件表达的少数算例外，其余算例（占总数的 90%以上）都以 M 文件或 MDL 文件形式刻录在光盘的 **\mfiles** 文件夹中。

本光盘 M 文件应在 MATLAB6.5 以上版本运行；涉及符号计算的应有 Symbolic Math Toobox 2.1.3 以上版本 适配；MDL 文件应有 SIMULINK5.0 以上版本适配。

对于其他版本，或较低版的工具包，有些文件的运行可能会失败。但只要对个别指令稍加修改就可，有关这方面的叙述可参见相应印刷版。

（1）光盘 M、MDL 文件的功用

- **提供可直接运作的 M 源码文件**

  只要有 MATLAB 环境，本光盘上的 M 文件就可以运行。它的适用条件比 \mbook 文件夹上的 DOC 文件宽松得多，也就是不管读者是否正确安装 Notebook，不管文件产生的是动画还是交互操作界面，它们都能在 MATLAB 环境中正确执行。

  每个算例文件都是完整的，可在 MATLAB 环境中直接运行的，所得结果与印刷版相对应。但出于运行方式不同的考虑，有些光盘 M 文件与印刷版文件指令可能会存在少许差别，目的是为把算例特征表现得更充分。

  此外，本光盘提供的 M 文件中，有许多是很通用的，读者只要稍加修改，就可为己所用。
- **弥补了印刷版没有 SIMULINK 模型文件的缺陷**

  由于 SIMULINK 工作特点的缘故，所以迄今为止所有涉及 SIMULINK 的印刷版书籍中都没有能直接运行的模型文件。这给读者带来许多困惑和麻烦：一，读者如想

验证书中结论，那就不得不从建模做起；二，仿真模块中的参数设置常使初学者顾此失彼，而造成仿真失败。本光盘上 MDL 模型文件都可直接在 MATLAB 中运行，进行验证。用户也可以在模型打开后，修改参数，观察变化。

（2）mfiles 文件夹上文件的放置规则

- **exm 为前缀的文件都是可直接运行的算例文件**

  前缀后的编号与算例编号对应。最左边的两位数字为“章”编号标注，其后的数字是“节”编号。具体举例如下：

  【例 2.5.3-2】对应的 M 文件是 exm02053_2.m ；
  【例 5.8.3.5-3】对应的 M 文件是 exm050835_3.m ；
  【例 5.13.2.3-2】对应的 M 文件是 exm051323_2.m ；
  【例 9.3.2.1-1】对应的 MDL 文件是 exm09321_1.mdl ；
  【例 10.7.2-1】对应的是 exm10072_1.m 。

- **第 8 章中的 M、MDL 文件**

  第 8 章算例中存在同一个例题对应着 2 个同编号文件的情况。如【例 8.8.3.2-1】就有 exm080832m_1.m 和 exm080832_1.mdl 两个文件。后者是该例的 SIMULINK 模型文件，而前者是与此例配套的 M 文件。使用时，要注意文件名上的微小差异。

- **其他非 exm 前缀文件是被调用文件**

  在 \mfiles 文件夹上还有一些不以 exm 为前缀的文件，它们不与算例直接对应，而是必不可少的被调用文件。在印刷版上可以找到有关它们的说明。

- **mfiles 的三个子文件夹**

  子文件夹 \mfiles\@queue　　存放着定义“队列”对象方法的重载文件。
  子文件夹 \mfiles\@stack　　存放着定义“堆栈”对象方法的重载文件。
  子文件夹 \mfiles\private　　存放着只能被 mfiles 夹上函数调用的函数文件。

（3）M、MDL 的使用方法

直接在 MATLAB 指令窗中，运行（不带扩展名的）算例 M 文件名，就可得到相关结果。在此要再次提醒的是：必须把 \mfiles 文件夹设置在 MATLAB 的搜索路径上 。

B.6 matlab_c 文件夹上各种文件的使用

基于第 11 章的特殊性，其配套软件文件也与其他章节不同：一，为第 11 章专设一个文件夹 **\matlab_c** ；二，该文件夹上不仅包含算例文件，而且包含该章叙述内容中用到的文件；三，文件夹不仅存放着“进行编译操作的 M 文件”，而且存放着“被编译的 M（或 C）文件”和“编译得的 DLL（或 EXE）文件”。

（1）matlab_c 文件夹上文件的功用

- **弥补印刷版不能提供完整算例文件的缺陷**

  不管编译生成的是 DLL 文件，还是 EXE 文件，印刷版只能提供它们的运行结果，而不能给出文件本身。于是，读者也就无法亲自运作算例生成的目标文件。

  本文件夹可以完全克服印刷版的这一缺陷。

- **提供了读者练习 MATLAB 编译器所需的全部文件**

  编译能否成功受被编译文件的正确性、MATLAB 编译器能力、编译操作的正确性等诸多因素影响。本文件夹提供的算例文件可减轻读者挫折，帮助读者一个环节一个环节地克服练习中所遇到的困难。

（2）该文件夹命名规则

- **进行编辑操作的 M 文件以 exm 为前缀，其后紧跟算例编号或节次编号**

  例如：
  【例 12.1.4-1】对应的“编译操作 M 文件”是 exm12014_1.m ；
  “第 12.2.2.2 节（1）mex 应用程序的验证”所对应的“编译操作 M 文件”是 exm120222_1.m ；

- **被编译文件和编译所得文件的名称与印刷版一致**

（3）使用方法

- **“编辑操作 M 文件”的使用方法**
  每个文件都可以在 MATLAB 指令窗中直接运行。请注意：一，操作前必须先把 **\matlab_c** 设置在 MATLAB 的搜索路径上；二，操作后生成的目标文件被存放在 **D:\mywork** 文件夹中（假如该夹不存在，在操作过程中会自动创建）。
- **“被编译文件”的使用**
  本文件夹上的“被编译（M 或 C）文件”是专供读者学习第 11 章时使用的。它们的存在可避免因“被编程序自身的不适当”而导致编译失败。
- **“编译所得文件”的使用**
  本文件夹上的“编译所得的（DLL 或 EXE）文件”是专供读者对照验证用的。
  对于 DLL 文件来说，只要在 MATLAB 指令窗中输入相应文件名就可运行，并给出结果。
  但对于 EXE 文件来说，假如读者想在 DOS 环境中运行 \matlab_c 文件夹上的 EXE 文件，那么必须注意：或把 \matlab_c 设置为 DOS 环境下的当前文件夹，或使运行文件带完整的路径。

B.7 matlab_api 文件夹上各种文件的使用

该文件夹上的文件是供读者实践第 12 章内容使用的。该文件加上文件类型较多，各种文件所需的软件环境很不相同。为保证运行成功，请读者仔细阅读第 12 章的相关内容。

该文件夹上的文件编号原则与全书一致，与第 12 章相应算例编号对应。

象运行 matlab_c 文件夹上的文件一样，在运行 matlab_api 文件夹文件时，要特别注意所运作文件是否在当前目录或搜索路径上。

B.8 PowerMatlab 文件夹上的 PPT 文件的使用

该文件夹上存放着 3 个 PPT 文件及其附属文件。要运行这三个 PPT 文件，必须在机器上事先安装 PowerPoint 。

（1）PPT 文件的功用

- **弥补印刷版无法提供 PowerPoint 源码文件的缺点**
  本书第 13.4 节介绍的科技演讲稿涉及三个不同软件：PowerPoint，Word，MATLAB 之间的链接。算例 13.4.1-1、13.4.2-1 和 13.4.3-1 虽对 PPT 文件的制作进行了比较详细地介绍，但无法以习惯的文字方式向读者提供 PowerPoint 源码文件。这无疑给读者演练这两个算例带去困难。
  光盘上的 E_book.ppt，exm13042_1.ppt，exm13043_1.ppt 文件可以直接运行，播放幻灯，并导出相应的交互界面。
- **提供制作科技演讲稿的样本**
  光盘上提供的 3 个 PPT 文件具有典型性。E_book.ppt 可以在幻灯演讲中进行多章节切换并引入复杂的现场计算；exm13042_1.ppt 则可以从幻灯演讲中直接导出进行科学计算的交互界面；exm13043_1.ppt 则可以幻灯放映过程中现场进行科学计算。

（2）PPT 文件的使用方法

关于 E_book.ppt、exm13042_1.ppt 和 exm13043_1.ppt 的使用方法，在印刷版第 13.4.1、13.4.2、13.4.3 节有详细的叙述。

B.9 其他

在本书印刷版发行后，光盘软件的更新内容将通过光盘上所带的 Readme 文件发布。

参 考 文 献

[1] M.L.Abell, J.P.Braselton, Maple Ⅴ by Example, Academic Press , 1999 .
[2] S.J.Chapman, MATLAB Programming for Engineers, Brooks/Cole, CA, 2002.
[3] D.Hanselman, B.Littlefield, Mastering MATLAB 5, Prentice Hall, New Jersey, 1998.
[4] G.J.Borse, Numerical Methods with MATLAB, PWS, Boston, 1997.
[5] L.W.Couch Ⅱ, Digital and Communication Systems , Prentice Hall , New Jersey , 1997 .
[6] R.C.Dorf, R.H.Bishop, Modern Control Systems, Addison-Wesley Publishing Company, England, 2001.
[7] G.F.Franklin, J.D.Powell, A.Emami-Naeini, Feedback Control of Dynamic Systems, Prentice Hall, New Jersey, 2002.
[8] V.K.Ingle, J.G.Proakis, Digital Signal Processing using MATLAB, PWS, Boston, 2000.
[9] MathWorks, MATLAB 6.0, 2000.
[10] MathWorks, MATLAB 6.1, 2001.
[11] MathWorks, MATLAB 6.5, 2002.